Université Paris-Dauphine
École Doctorale Décision, Informatique, Mathématique, Organisation
Centre de Recherche en Mathématiques de la Décision

Thèse présentée par
Christian André SCHÄFER

Pour obtenir le grade de
Docteur en Science

Spécialité
Statistique Mathématique

---

# Monte Carlo methods for sampling high-dimensional binary vectors

---

Thèse dirigée par Nicolas CHOPIN
Soutenue le 14/11/2012

Jury composé de:

| | | |
|---|---|---|
| M. Nicolas CHOPIN | ENSAE-CREST | Directeur de thèse |
| M. Chris HOLMES | University of Oxford | Rapporteur |
| M. Jean-Michel MARIN | Université Montpellier II | Rapporteur |
| Mme Gersende FORT | Télécom ParisTech | Examinatrice |
| M. Christian ROBERT | Université Paris-Dauphine | Examinateur |

# Acknowledgements

First of all, I would like to thank my PhD advisor Nicolas Chopin for the patience, the encouragement and the time he sacrificed in discussions and proof-reading manuscripts. I am pleased to be the first in a certainly fast-growing list of PhD students he will supervise in his academic career. Further, I would like to thank the CREST for financing my thesis and allowing me to solely concentrate on the research without any further obligations.

I was fortunate to spend a week at the Biometris Laboratory in Wageningen to learn a little bit about the variable selection problems which arise in the context of plant breeding. I would like to thank former CREST postdoc Willem Kruijer for his invitation and the warm welcome in Wageningen. The stay was financed by the European Cooperation in Science and Technology. I would further like to thank Omiros Papaspiliopoulos for organizing a great six-week visit to Universitat Pompeu Fabra in Barcelona and introducing me to his colleagues. The stay was financed by the Fondation Sciences Mathématiques de Paris.

Finally, I would like to thank my PhD colleagues Celine Duval and Pierre Jacob for the good times we had in the three years we shared the F14 office in the unbeloved annex facing the INSEE tower. Together with my PhD colleages Julyan Arbel, Guillaume Lepage and former CREST postdoc Robin Ryder, they taught me most of what I now know about France. I would particularly like to thank them all for their patience and their effort to improve my French. I keep great memories of conferences we went to together and the many after-work-beers we shared with PhD colleague Giuseppe Benedetti in the finest taverns of Malakoff.

Last, but not least, I want to thank Kerstin Steinberg for her enduring support and encouragement throughout my thesis.

# Summary

This thesis is concerned with Monte Carlo methods for sampling high-dimensional binary vectors from complex distributions of interest. If the state space is too large for exhaustive enumeration, these methods provide a mean of estimating the expected value with respect to some function of interest. Standard approaches are mostly based on random walk type Markov chain Monte Carlo, where the equilibrium distribution of the chain is the target distribution and its ergodic mean converges to the expected value. While these methods are well-studied and asymptotically valid, convergence of the Markov chain might be very slow if the target distribution is highly multi-modal. We propose a novel sampling algorithm based on sequential Monte Carlo methodology which copes well with multi-modal problems by virtue of an annealing schedule. The usefulness of this approach is demonstrated in the context of Bayesian variable selection and combinatorial optimization of pseudo-Boolean objective functions.

**Chapter 1** The introductory section provides an overview of existing Monte Carlo techniques for sampling from binary distributions and particularly reviews the standard Markov chain Monte Carlo methodology which is frequently used in practice. We introduce the notion of multi-modality and discuss why random walk type Markov chains might fail to converge in a reasonable amount of time due to strong dependencies in the distribution of interest. This motivates the work on novel Monte Carlo algorithms which are more robust against multi-modality but still scale to high dimensions.

**Chapter 2** We describe a sequential Monte Carlo approach as an alternative sampling scheme which propagates a system of particles from an easy initial distribution, via intermediate instrumental distributions towards the distribution of interest. While the resample-move methodology comes from the standard toolbox of particle filtering (Del Moral et al., 2006), the central innovation is the use of a Metropolis-Hastings kernel with independent proposals in the move step of the algorithm. We achieve high acceptance rates and thus very fast mixing owing to advanced parametric families which efficiently approximate the intermediate distributions.

**Chapter 3** The performance of the proposed sequential Monte Carlo sampler depends on the ability to sample proposals from auxiliary distributions which are, in a certain sense, close to the current distribution of interest. This chapter contains the core work of this thesis and elaborates on strategies to construct parametric families for sampling binary vectors with dependencies. We work out practical solutions which can be incorporated in particle algorithms on binary spaces but also discuss approaches to modeling random binary vectors which are beyond the immediate Monte Carlo application. The practical scope of the proposed parametric families is examined in a numerical study on random cross-moment matrices.

**Chapter 4** The major statistical application for sampling binary vectors is Bayesian variable selection for linear regression models where quantities like the posterior inclusion probabilities of the predictors need to be computed. This chapter provides a brief introduction to variable selection in the context of normal linear models, where the posterior distribution is available in closed-form for a judicious choice of prior distributions on the model parameters. We construct several challenging test instances from real data, chosen to be considerably multi-modal, and compare the performance of the sequential Monte Carlo sampler to standard Markov chain Monte Carlo methods (George and McCulloch, 1997).

**Chapter 5** This chapter deals with ideas to extend the sequential Monte Carlo methodology to Bayesian variable selection in the context of generalized linear models with binary response like logistic or probit regression models. In this case, the posterior distribution is not available in closed-form, and the model parameters need to be integrated out using either approximations or pseudo-marginal ideas in order to apply the sequential Monte Carlo framework. Analogously to Chapter 4, we construct several test instances from real data and compare the performance of the sequential Monte Carlo sampler to the automatic generic sampler (Green, 2003) which is a trans-dimensional Markov chain Monte Carlo sampling scheme.

**Chapter 6** Stochastic optimization of pseudo-Boolean objective functions is a field of major interest in operations research since many important NP-hard combinatorial problems can be formulated in terms of binary programming. If the objective function is multi-modal, local search algorithms often fail to detect the global optimum and particle driven methods may provide more robust results. We discuss how the sequential Monte Carlo sampler can be used in an optimization context and show how the cross-entropy method by Rubinstein (1997) can be embedded in the sequential Monte Carlo framework.

In numerical experiments, we show that the parametric families proposed in Chapter 3 tremendously improve the performance of the cross-entropy method and compare the particle driven optimization schemes to local search algorithms.

**Chapter 7** We present some final remarks concerning particle algorithms on binary state spaces and points out some interesting lines for further research.

# Resumé

Cette thèse est consacrée à l'étude des méthodes de Monte Carlo pour l'échantillonnage de vecteurs binaires de grande dimension à partir de lois cibles complexes. Si l'espace-état est trop grand pour une énumération exhaustive, ces méthodes permettent d'estimer l'espérance d'une loi donnée par rapport à une fonction d'intérêt. Les approches standards sont principalement basées sur les méthodes Monte Carlo à chaîne de Markov de type marche aléatoire, où la loi stationnaire de la chaîne est la distribution d'intérêt et la moyenne de la trajectoire converge vers l'espérance par le théorème ergodique. Bien que ces méthodes soient bien étudiées et asymptotiquement valides, la convergence de la chaîne de Markov peut être très lente si la loi cible est fortement multimodale. Nous proposons un nouvel algorithme d'échantillonnage basé sur les méthodes de Monte Carlo séquentielles qui sont plus robustes au problème de multimodalité grâce à une étape de recuit simulé. L'utilité de cette approche est démontrée dans le cadre de sélection bayésienne de variables et l'optimisation combinatoire des fonctions pseudo-booléennes.

**Chapitre 1** Cette section introductive donne un aperçu des techniques existantes de Monte Carlo pour l'échantillonnage de vecteurs binaires. On y examine notamment les méthodes de Monte Carlo à chaîne de Markov qui sont fréquemment utilisées dans la pratique. La notion de multimodalité y est introduite, suivie d'une discussion sur les chaînes de Markov de type marche aléatoire qui souvent ne convergent pas en un temps computationnel raisonnable, en raison des fortes dépendances parmi les composantes de la loi d'intérêt, ce qui motive le développement de nouveaux algorithmes de type Monte Carlo qui soient plus robustes face à la multimodalité mais aussi utilisables en grande dimension.

**Chapitre 2** Nous proposons une technique d'échantillonnage alternative basée sur les méthodes de Monte-Carlo séquentielles qui propage un système de particules à partir d'une loi initiale simple, par des lois intermédiaires auxiliaires vers la loi cible. Alors que la méthodologie *resample-move* provient de la boîte à outils standard du filtrage particulaire (Del Moral et al., 2006), l'innovation centrale est l'utilisation d'un noyau de

Metropolis-Hastings avec des propositions indépendantes dans l'étape de déplacement. L'usage des familles paramétriques avancées qui approchent efficacement les lois intermédiaires et permettent d'atteindre des taux d'acceptation élevés nécessaires pour la construction de chaînes de Markov rapidement mélangeantes.

**Chapitre 3** La performance de l'échantillonneur de Monte Carlo séquentiel dépend de la capacité d'échantillonner selon des lois auxiliaires qui sont, en un certain sens, proche à la loi de l'intérêt. Ce chapitre contient le travail principal de cette thèse et présente des stratégies visant à construire des familles paramétriques pour l'échantillonnage de vecteurs binaires avec dépendances. Nous proposons des solutions pratiques qui peuvent être incorporées dans les algorithmes particulaires sur les espaces binaires, mais aussi des approches de modélisation de vecteurs binaires aléatoires qui sont au-delà de l'application immédiate de méthodes Monte-Carlo. L'intérêt pratique des familles paramétriques proposées est examiné dans une étude numérique sur des matrices aléatoires de moments croisés.

**Chapitre 4** L'application statistique majeure pour d'échantillonnage de vecteurs binaires est la sélection bayésienne de variables parmi des modèles de régression linéaire où des quantités telles que les probabilités d'inclusion a posteriori des prédicteurs doivent être calculées. Ce chapitre propose une brève introduction à la sélection de variables dans le cadre de modèles linéaires normaux, où la distribution a posteriori est disponible sous forme analytique pour un choix judicieux de la loi a priori sur les paramètres du modèle. Nous construisons plusieurs instances de test exigeants sur données réelles, choisis pour être considérablement multimodal, et l'échantillonneur de Monte Carlo séquentiel est comparé avec des méthodes standards de Monte Carlo à chaîne de Markov (George and McCulloch, 1997).

**Chapitre 5** Ce chapitre propose des idées pour étendre les méthodes de Monte Carlo séquentielles à la sélection bayésienne de variables dans le contexte des modèles linéaires généralisés à réponse binaire comme les modèles de régression logistique ou probit. Dans ce cas, la distribution a posteriori n'est pas disponible sous forme fermée, et les paramètres du modèle doivent être marginalisés à l'aide soit d'approximations, soit d'approches pseudo-marginales afin d'appliquer l'algorithme de Monte Carlo séquentiel. Par analogie au chapitre 4, plusieurs instances de test sur données réelles sont construites et l'échantillonneur de Monte Carlo séquentiel est comparé à l'échantillonneur automatique générique (Green, 2003) qui est une méthode de Monte Carlo à chaîne de Markov transdimensionnel.

**Chapitre 6** L'optimisation stochastique de fonctions pseudo-booléennes est un domaine d'intérêt majeur en recherche opérationnelle car des nombreuses problèmes combinatoires NP-complet peuvent être formulés en termes de programmation binaire. Si la fonction objective est multimodale, les algorithmes de recherche locale ne parviennent souvent pas à détecter l'optimum global et les méthodes particulaires peuvent donner des résultats plus robustes. Nous détaillons comment l'échantillonneur de Monte Carlo séquentiel peut être utilisé dans un contexte d'optimisation et comment la méthode de l'entropie croisée de Rubinstein (1997) peut être intégré dans le cadre de l'algorithme Monte Carlo séquentiel. Les expériences numériques montrent que les familles paramétriques proposées dans le chapitre 3 améliorent considérablement la performance de la méthode de l'entropie croisée. Finalement, les méthodes particulaires sont comparées aux algorithmes de recherche locale.

**Chapitre 7** La conclusion de cette thèse présente quelques remarques finales concernant les algorithmes particulaires sur les espaces d'états binaires et des perspectives de recherche pour intégrer les familles paramétriques dans d'autres applications.

# Contents


**I. Methodology** 17

**1. Introduction to sampling random binary vectors** 19

1.1. Introduction . . . 19
1.1.1. Notation . . . 20
1.1.2. Importance sampling . . . 21
1.2. Markov chain Monte Carlo . . . 22
1.2.1. Markov chain Monte Carlo estimators . . . 23
1.2.2. Normalized estimators . . . 24
1.3. The Metropolis-Hastings kernel . . . 25
1.3.1. Random walk kernels . . . 25
1.3.2. Metropolis-Hastings independence sampler . . . 28
1.4. Adaptive Markov chain Monte Carlo . . . 29
1.4.1. Adaptive metropolized Gibbs . . . 30
1.4.2. Adaptive random walk . . . 31
1.4.3. Adaptive independence sampler . . . 32
1.5. Multi-modality . . . 32
1.5.1. Markov chains and multi-modality . . . 32
1.5.2. Bayesian adaptive sampling . . . 33

**2. The sequential Monte Carlo sampler** 35

2.1. Introduction . . . 35
2.2. Sequential Importance Sampling . . . 36
2.2.1. Importance weights . . . 37
2.2.2. Optimal step length . . . 37
2.2.3. Resampling step . . . 38
2.3. Adaptive move step . . . 39
2.3.1. Fast-mixing kernels . . . 39
2.3.2. Adaptive stopping rule . . . 40

2.4. Remark on discrete state spaces . . . 41
2.4.1. Impact on the effective sample size . . . 41
2.4.2. Impact on the resample-move step . . . 41

**3. Parametric families on binary spaces 43**
3.1. Motivation . . . 43
3.1.1. Product family . . . 44
3.1.2. Logistic conditionals family . . . 45
3.1.3. Gaussian copula family . . . 45
3.2. Preliminaries on random binary vectors . . . 46
3.2.1. Cross-moments and correlations . . . 46
3.2.2. Representations and bounds . . . 48
3.3. Families based on generalized linear models . . . 50
3.3.1. Definition . . . 50
3.3.2. Maximum-likelihood . . . 53
3.3.3. Method of moments . . . 55
3.4. Families based on multivariate copulas . . . 56
3.4.1. Definition . . . 56
3.4.2. Further copula approaches . . . 57
3.4.3. Method of moments . . . 57
3.5. Families based on other techniques . . . 58
3.5.1. Multiplicative interactions . . . 58
3.5.2. Additive interactions . . . 62
3.6. Practical scope . . . 67
3.6.1. Sparse families . . . 67
3.6.2. Random cross-moment matrices . . . 69
3.6.3. Computational results . . . 70
3.6.4. Discussion . . . 71

**II. Applications 73**

**4. Bayesian variable selection for normal linear models 75**
4.1. Introduction . . . 75
4.1.1. Selection criteria . . . 76
4.1.2. Bayesian variable selection . . . 77
4.1.3. Penalized likelihood criteria . . . 77
4.1.4. Convex optimization . . . 78

4.2. Marginal likelihood . . . 78
4.2.1. Hierarchical priors . . . 79
4.2.2. Zellner's prior . . . 79
4.2.3. Independent prior . . . 80
4.3. Priors on the model space . . . 80
4.3.1. Prior on the model size . . . 80
4.3.2. Main effect restrictions . . . 81
4.4. Sequential Monte Carlo . . . 81
4.4.1. Intermediate distributions . . . 81
4.4.2. Parametric families . . . 82
4.5. Numerical experiments . . . 84
4.5.1. Construction of test instances . . . 85
4.5.2. Comparison and conclusion . . . 87
4.5.3. Assets and drawbacks . . . 89

**5. Bayesian variable selection for binary response models 97**
5.1. Introduction . . . 97
5.1.1. Selection criteria . . . 98
5.1.2. Bayesian variable selection . . . 98
5.2. Marginal likelihood . . . 99
5.2.1. Maximum likelihood . . . 99
5.2.2. Prior on the regression parameters . . . 101
5.2.3. Laplace approximation . . . 101
5.2.4. Pseudo-marginal sampler . . . 101
5.2.5. Corrected Laplace sampler . . . 103
5.3. Transdimensional Markov chain Monte Carlo . . . 103
5.3.1. Reversible jumps . . . 103
5.3.2. The automatic generic sampler . . . 104
5.4. Numerical experiments . . . 104
5.4.1. Construction of test instances . . . 105
5.4.2. Comparison and conclusion . . . 107

**6. Pseudo-Boolean optimization 111**
6.1. Introduction . . . 111
6.1.1. Statistical modeling . . . 113
6.1.2. Rare event simulation . . . 114
6.2. Optimization algorithms . . . 115
6.2.1. Sequential Monte Carlo . . . 115

6.2.2. Cross-entropy method . . . 116
6.2.3. Simulated annealing . . . 117
6.2.4. Randomized local search . . . 118
6.3. Application . . . 119
6.3.1. Unconstrained Quadratic Binary Optimization . . . 119
6.3.2. Particle optimization and meta-heuristics . . . 120
6.3.3. Particle optimization and exact solvers . . . 121
6.3.4. Construction of test problems . . . 121
6.4. Numerical experiments . . . 125
6.4.1. Toy example . . . 125
6.4.2. Random test instances . . . 126
6.4.3. Comparison of binary parametric families . . . 127
6.4.4. Comparison of optimization algorithms . . . 127
6.5. Discussion and conclusion . . . 129

**7. Conclusion and outlook** **133**
7.1. The independence sampler . . . 133
7.2. Scaling to higher dimensions . . . 134
7.3. Parallel computing . . . 134

**Software** **137**

**Glossary** **139**

**Acronyms** **141**

# Part I.

# Methodology

# 1. Introduction to sampling random binary vectors

## Resumé

*Cette section introductive donne un aperçu des techniques existantes de Monte Carlo pour l'échantillonnage de vecteurs binaires. On y examine notamment les méthodes de Monte Carlo à chaîne de Markov qui sont fréquemment utilisées dans la pratique. La notion de multimodalité y est introduite, suivie d'une discussion sur les chaînes de Markov de type marche aléatoire qui souvent ne convergent pas en un temps computationnel raisonnable, en raison des fortes dépendances parmi les composantes de la loi d'intérêt, ce qui motive le développement de nouveaux algorithmes de type Monte Carlo qui soient plus robustes face à la multimodalité mais aussi utilisables en grande dimension.*

## 1.1. Introduction

In this chapter, we review standard Monte Carlo methods for sampling high-dimensional binary vectors and motivate the work on an alternative sampling scheme based on sequential Monte Carlo (SMC) methodology. Most of this discussion was published in Schäfer and Chopin (2012). Standard approaches are typically based on random walk type Markov chain Monte Carlo (MCMC), where the equilibrium distribution of the chain is the distribution of interest and its ergodic mean converges to the expected value of interest. While MCMC methods are asymptotically valid, convergence of Markov chains may be very slow if the distribution of interest is highly multi-modal.

In Chapter 2, we propose a novel algorithm based on SMC methodology which copes well with multi-modal problems by virtue of an annealing schedule. This work approaches a well-studied problem from a different angle and provides new perspectives. Firstly, there is numerical evidence that particle methods, which track a population of particles, initially well spread over the sampling space, are often more robust than local

methods based on MCMC, since the latter are prone to get trapped in the neighborhood of local modes. We largely illustrate this effect in our simulation studies in Chapters 4, 5 and 6. Secondly, SMC type algorithms are easily parallelizable, and parallel computing for Monte Carlo algorithms has gained a tremendous interest in the very recent years (Lee et al., 2010; Suchard et al., 2010), due to the increasing availability of multi-core processing units in standard computers.

Thirdly, we argue that the SMC sampler is fully adaptive and requires practically no tuning to perform well. A Monte Carlo algorithm is said to be adaptive if it adjusts, sequentially and automatically, its sampling distribution to the problem at hand. Important classes of adaptive Monte Carlo are sequential Monte Carlo (e.g. Del Moral et al., 2006), adaptive importance sampling (e.g. Cappé et al., 2008) and adaptive Markov chain Monte Carlo (e.g. Andrieu and Thoms, 2008), among others. The choice of the parametric family which defines the range of possible sampling distributions is critical for good performance. We address this question in Chapter 3.

### 1.1.1. Notation

Throughout this thesis, vectors are denoted in italic and matrices in straight bold-faced type. Sets, random variables and matrices are denoted by capital letters.

We write $\mathbb{B} := \{0, 1\}$ for the binary space. For $b \geq a$, we denote by $[\![a, b]\!] := \{x \in \mathbb{Z} \mid a \leq x \leq b\}$ the discrete and by $[a, b) := \{x \in \mathbb{R} \mid a \leq x < b\}$ the continuous interval. We denote by $d \in \mathbb{N}$ the generic dimension and $n \in \mathbb{N}$ the generic sample size and define the index sets $D := [\![1, d]\!]$ and $N := [\![1, n]\!]$ for ease of notation.

Let $\mathcal{P}(M)$ denote the power set and $\mathcal{B}(M)$ the Borel $\sigma$-field generated by the set $M$. Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. A random variable $X \colon \Omega \to \mathbb{X}$ is defined on a measurable space $(\mathbb{X}, \mathcal{X})$ which in our case is either $(\mathbb{B}^d, \mathcal{P}(\mathbb{B}^d))$ or $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ or countable products of these. We write $X \sim \mu$ if $\mu = \mathbb{P} \circ X^{-1}$ and say that $X$ has the distribution $\mu$. For a $\mu$-integrable function $f \colon \mathbb{X} \to \mathbb{R}$, we denote by

$$\mu(f) := \mathbb{E}_\mu\left(f(\boldsymbol{X})\right) := \int_{\mathbb{X}} f(\boldsymbol{x})\mu(d\boldsymbol{x})$$

the expected value of $f$ with respect to $\mu$; if $f$ is the identity mapping, we write $m^\mu$ for the mean. Since this work is mostly concerned with sampling from measures defined on the finite state space $\mathbb{B}^d$, some technical difficulties arising in general measure theory can be neglected. We do not distinguish between the probability measure $\mu \colon \mathcal{P}(\mathbb{B}^d) \to [0, 1]$ and its mass function $\pi \colon \mathbb{B}^d \to [0, 1]$, $\pi(\boldsymbol{\gamma}) = \mu(\{\boldsymbol{\gamma}\})$ but refer to both mappings by the same symbol $\pi$; we also write $\pi^n$ instead of $\mu^{\otimes n}$ for the $n$-fold product measure.

Generally, we let $\pi$ denote the binary distribution of interest, $(q_\theta)_{\theta\in\Theta}$ a parametric family of distributions and $\kappa$ a Markov transition kernel. In a Bayesian context, we let $\mathcal{L}$ denote the likelihood, $p$ the prior distribution and $\pi$ the posterior distribution, where the arguments of the mass functions usually indicate the context, that is, for example, $p(\theta) = \mathcal{N}(0,1)$ means that the parameter $\theta$ is a priori standard normal distributed.

### 1.1.2. Importance sampling

A non-zero mapping $\tilde{\pi}\colon \mathbb{B}^d \to [0,\infty)$ defines a probability measure $\pi \propto \tilde{\pi}$ on $(\mathbb{B}^d, \mathcal{P}(\mathbb{B}^d))$, where $\propto$ denotes equality up to a scaling factor. The goal is to sample from $\pi$ in order to approximate quantities like the expected value of $f\colon \mathbb{B}^d \to \mathbb{R}$

$$\pi(f) = \mathbb{E}_\pi\left(f(\boldsymbol{X})\right) = \sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} f(\boldsymbol{\gamma})\pi(\boldsymbol{\gamma}) = \frac{\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} f(\boldsymbol{\gamma})\tilde{\pi}(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d}\tilde{\pi}(\boldsymbol{\gamma})} \tag{1.1}$$

although the normalizing constant may be unknown. Even for moderate $d \in \mathbb{N}$, the state space is too large for exhaustive enumeration. In this case, one may resort to Monte Carlo methods to provide an estimate $\hat{\pi}(f)$ of the intractable quantity $\pi(f)$. If we can draw independent and indentically distributed (IID) samples $(\boldsymbol{X}_1,\dots,\boldsymbol{X}_n) \sim \pi^n$, we have an unbiased estimator

$$\hat{\pi}^n_{\text{IID}}(f) := n^{-1}\sum_{k=1}^n f(\boldsymbol{X}_k),$$

and $\hat{\pi}^n_{\text{IID}}(f) \overset{n\to\infty}{\longrightarrow} \pi(f)$ a.s. by virtue of the strong law of large numbers. Generally, however, we cannot draw independent samples from $\pi$. Let $q$ denote an *instrumental* or *auxiliary* distribution. For an independent sample $(\boldsymbol{X}_1,\dots,\boldsymbol{X}_n) \sim q^n$, we have an asymptotically unbiased importance sampling (IS) estimator

$$\hat{\pi}^n_{\text{IS}}(f) := \frac{\sum_{k=1}^n f(\boldsymbol{X}_k)w(\boldsymbol{X}_k)}{\sum_{k=1}^n w(\boldsymbol{X}_k)}$$

of the expected value where $w(\boldsymbol{\gamma}) := \tilde{\pi}(\boldsymbol{\gamma})/\tilde{q}(\boldsymbol{\gamma})$ where $\tilde{q} \propto q$. The ratios of the (not necessarily normalized) mass functions of the instrumental and the target distribution are referred to as *importance weights*. The instrumental distribution has to verify $\operatorname{supp}(\pi) \subseteq \operatorname{supp}(q)$ to ensure that $\hat{\pi}^n_{\text{IS}}(f) \overset{n\to\infty}{\longrightarrow} \pi(f)$ a.s. by virtue of the strong law of large numbers, see Robert and Casella (2004, sec. 3.3). The asymptotic variance of the estimator can roughly be approximated by

$$\mathbb{V}[\hat{\pi}^n_{\text{IS}}(f)] \approx \mathbb{V}_\pi[f(\boldsymbol{X}_1)]n^{-1}\left(1 + \mathbb{V}_q[w(\boldsymbol{X}_1)]/c^2\right),$$

where $c > 0$ is some unknown normalizing constant (Liu, 1996a; Kong et al., 1994). The last term on the right hand side can be estimated by

$$\hat{\eta}^{-1} := \frac{\sum_{k=1}^{n} w(\boldsymbol{x}_k)^2}{[\sum_{k=1}^{n} w(\boldsymbol{x}_k)]^2} \approx n^{-1} \left(1 + \mathbb{V}_q[w(\boldsymbol{X}_1)]/c^2\right) \tag{1.2}$$

where $\hat{\eta} \in [1, n]$ is the so-called effective sample size (ESS). Since $\hat{\eta}$ is an estimate for an approximation to an asymptotic quantity, it might be substantially misleading. However, the ESS is widely used in practice because it is easy to compute and does not depend on $f$. The name stems from the common interpretation that the precision of an IS estimator $\hat{\pi}^n_{\text{IS}}(f)$ is about the same as the precision of an IID estimator $\hat{\pi}^{\lfloor \hat{\eta} \rfloor}_{\text{IID}}(f)$.

The instrumental distribution which minimizes the variance of the importance sampling estimator is $q^* \propto |f(\cdot)|\, \tilde{\pi}$. Typically, we cannot generate independent samples from any distribution close to $q^*$ and have to rely on sub-optimal instrumental distributions which often yield extremely inefficient importance sampling estimators.

## 1.2. Markov chain Monte Carlo

We introduce some notation and review a few well-known results from Markov chain theory (see e.g. Meyn et al., 2009). A time-homogeneous Markov chain on the binary space is a sequence of random variables $(\boldsymbol{X}_k)_{k \in N_0} \sim (p\kappa^n)$ which enjoys the Markov property and is completely defined by its *initial distribution* $p$ and its *transition kernel* $\kappa$, that is

$$\mathbb{P}\left(\boldsymbol{X}_0 = \boldsymbol{x}_0, \ldots, \boldsymbol{X}_n = \boldsymbol{x}_n\right) = p(\boldsymbol{x}_0) \prod_{k=1}^{n} \kappa(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}).$$

We denote by $(p\kappa^n)$ the mass function of a chain up to time $n \in \mathbb{N}$ and by $[p\kappa^n]$ the *marginal distribution* of the chain at time $n \in \mathbb{N}$ which is obtained by repeated application of the *transition operator*

$$[p\kappa] := \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} p(\boldsymbol{\gamma}) \kappa(\cdot \mid \boldsymbol{\gamma}).$$

In the sequel, we only consider *aperiodic* Markov chains which are *irreducible* and therefore *positive recurrent* on a finite state space. Then the transition operator has a *unique* fixed point

$$[\pi\kappa] = \pi \tag{1.3}$$

referred to as the *invariant* or *equilibrium* distribution. The Markov chain is *stationary* if and only if $p = \pi$. On finite spaces, the total variation (TV) norm of the measure $\pi$

is given by $\|\pi\|_{\text{TV}} := \frac{1}{2}\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} |\pi(\boldsymbol{\gamma})|$. The total variation distance between the marginal and the equilibrium distribution of the Markov chain is bounded by

$$\|[p\kappa^n] - \pi\|_{\text{TV}} \leq \lambda_2^n \, c(p) \tag{1.4}$$

where $\lambda_2$ is the second-largest eigenvalue of the kernel and $c(p) > 0$ a constant depending on the initial distribution. Note that $\lambda_2 < 1$ since the Markov chain is aperiodic. For a Markov chain to admit $\pi$ as its unique equilibrium distribution distribution, it is sufficient that for all $\boldsymbol{x}, \boldsymbol{\gamma} \in \mathbb{B}^d$

$$\pi(\boldsymbol{x})\kappa\left(\boldsymbol{\gamma} \mid \boldsymbol{x}\right) = \pi(\boldsymbol{\gamma})\kappa\left(\boldsymbol{x} \mid \boldsymbol{\gamma}\right). \tag{1.5}$$

Equation (1.5) is also referred to as *detailed balance* condition and a Markov chain with detailed balance is said to be *reversible* with respect to $\pi$.

A positive recurrent, irreducible and aperiodic Markov chain is *ergodic* (Robert and Casella, 2004) which means that the measure-preserving dynamical system defined by the probability space and the shift operator on the stationary Markov chain yields the same quantities when averaged over the states visited by the chain as when averaged over all states of the state space weighted according to their probabilities. Let $(\boldsymbol{X}_k)_{k\in N_0} \sim (\pi\kappa^n)$ be an ergodic Markov chain and $f\colon \mathbb{B}^d \to \mathbb{R}$ a function. From the ergodic theorem, it follows that

$$(n+1)^{-1}\sum_{k=0}^{n} f(\boldsymbol{X}_k) \overset{n\to\infty}{\longrightarrow} \pi(f) \quad \text{a.s.},$$

which generalizes the strong law of large numbers to random variables with Markovian dependencies.

### 1.2.1. Markov chain Monte Carlo estimators

The idea of MCMC is to construct a transition kernel $\kappa$ which admits the distribution of interest $\pi$ as unique equilibrium distribution. If we can sample a Markov chain $(\boldsymbol{X}_0, \ldots, \boldsymbol{X}_n) \sim (\pi\kappa^n)$, we have an unbiased estimator

$$\hat{\pi}^n_{\text{MCMC}}(f) := (n+1)^{-1}\sum_{k=0}^{n} f(\boldsymbol{X}_k)$$

by virtue of the ergodic theorem for Markov chains. Typically, we cannot provide an initial draw from the distribution of interest $\pi$ since in this case we would prefer to construct an estimator $\hat{\pi}^n_{\text{IID}}$ based on independent samples. For a different initial distribution $p \neq \pi$, the Markov chain $(\boldsymbol{X}_0, \ldots, \boldsymbol{X}_n) \sim (p\kappa^n)$ is not stationary but (1.4) ensures

that the equilibrium distribution is approximately obtained after $b \in \mathbb{N}$ steps. The first $b$ samples are then discarded as so-called burn-in period and the MCMC estimator becomes

$$\hat{\pi}^n_{\text{MCMC}}(f) := n^{-1} \sum_{k=b}^{n+b} f(\boldsymbol{X}_k).$$

The MCMC estimator is justified by asymptotic arguments. However, in practice it is often hard to guarantee that the stationary distribution is indeed approximately reached after $b$ steps and that the sampled trajectory is indeed approximately ergodic after $n$ steps. How large we have to choose $b$ and $n$ to ensure a desired precision of the Monte Carlo estimate depends on the mixing properties of the Markov kernel, that is the dependence on the past of the trajectory.

### 1.2.2. Normalized estimators

Some authors (Clyde et al., 2011) argue that the equilibrium sampling approach using MCMC might be sub-optimal on a large discrete state space, since the number of repeated visits to a state is mostly zero or small and therefore a poor estimator of the frequency. Consider the following improved estimator. Let $(\boldsymbol{x}_0, \dots, \boldsymbol{x}_n)$ denote a sample and

$$n(\boldsymbol{\gamma}) = \sum_{k=0}^{n} \delta_{\boldsymbol{x}_k}(\boldsymbol{\gamma})$$

the number of times the vector $\boldsymbol{\gamma}$ is in the sample. Further, let $V = \{\boldsymbol{x}_0, \dots, \boldsymbol{x}_n\}$ denote the set of all vectors which where sampled. The MCMC estimator (1.2.1) can be written

$$\hat{\pi}^n_{\text{MCMC}}(f) = \sum_{\boldsymbol{\gamma} \in V} f(\boldsymbol{\gamma}) \frac{n(\boldsymbol{\gamma})}{n+1}$$

where the frequencies $n(\boldsymbol{\gamma})/(n+1)$ are estimates of the probabilities $\pi(\boldsymbol{\gamma}) \propto \tilde{\pi}(\boldsymbol{\gamma})$ for all $\boldsymbol{\gamma} \in V$. We might therefore replace the estimated frequencies by their true values, which looks somewhat like an IS estimator

$$\hat{\pi}^n_{\text{IS}*}(f) = \sum_{\boldsymbol{\gamma} \in V} f(\boldsymbol{\gamma}) \frac{\tilde{\pi}(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma} \in V} \tilde{\pi}(\boldsymbol{\gamma})} \tag{1.6}$$

with importance function $\tilde{\pi}$. Although biased, this estimator might even be more efficient than the original one due to a Rao-Blackwellization effect. This raises the question whether equilibrium sampling using MCMC methodology is the adequate approach for sampling on binary space at all; García-Donato and Martínez-Beneito (2011) provide an interesting discussion and numerical experiments to investigate the merits of normalized estimators.

## 1.3. The Metropolis-Hastings kernel

Most transition kernels used in MCMC are some variant of the Metropolis-Hastings kernel,

$$\kappa_q(\boldsymbol{\gamma} \mid \boldsymbol{x}) := \alpha_q(\boldsymbol{\gamma}, \boldsymbol{x}) q(\boldsymbol{\gamma} \mid \boldsymbol{x}) + \delta_{\boldsymbol{x}}(\boldsymbol{\gamma}) \left[1 - \sum_{\boldsymbol{y} \in \mathbb{B}^d} \alpha_q(\boldsymbol{y}, \boldsymbol{x}) q(\boldsymbol{y} \mid \boldsymbol{x})\right]$$

where $q(\boldsymbol{\gamma} \mid \boldsymbol{x})$ is an *auxiliary* or *proposal* kernel and

$$\alpha_q(\boldsymbol{\gamma}, \boldsymbol{x}) := 1 \wedge \frac{\pi(\boldsymbol{\gamma}) q(\boldsymbol{x} \mid \boldsymbol{\gamma})}{\pi(\boldsymbol{x}) q(\boldsymbol{\gamma} \mid \boldsymbol{x})} \tag{1.7}$$

the *Metropolis-Hastings ratio* or *acceptance probability*. Obviously, it suffices to know the mass functions of $\pi$ and $q$ up to a constant, since the unknown normalizing constants cancel out in the Metropolis-Hastings ratio (1.7).

The name "acceptance probability" stems from the sampling procedure: The transition to the proposal state $\boldsymbol{Y} \sim q(\cdot \mid \boldsymbol{x})$ is accepted with probability $\alpha_q(\boldsymbol{Y}, \boldsymbol{x})$; the chain remains at the current state otherwise. The Metropolis-Hastings kernel verifies the detailed balance condition (1.5) and a proposal kernel with $\operatorname{supp}(\pi) \subseteq \operatorname{supp}([\delta_{\boldsymbol{x}} q^n])$ for all $n > n_0 \in \mathbb{N}$ and $\boldsymbol{x} \in \mathbb{B}^d$ ensures that the Markov chain is irreducible.

On discrete spaces accepting a proposal state does not necessarily imply that the state of the chain changes since the current state might have been proposed again. We distinguish between the acceptance probability (1.7) and the average mutation probability

$$\mu_q(\boldsymbol{x}) := \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d \setminus \{\boldsymbol{x}\}} \kappa(\boldsymbol{\gamma} \mid \boldsymbol{x}), \tag{1.8}$$

since high acceptance probabilities alone do not indicate good mixing. This is particularly true for random walk kernels on sampling problems with many local modes, as we demonstrate in the numerical experiments in Chapters 4 and 6.

### 1.3.1. Random walk kernels

We review some of the Markov transition kernels typically used for MCMC on binary spaces; this discussion has partially been published in Schäfer and Chopin (2012). Many popular Metropolis-Hastings kernels on binary spaces perform a random walk, that is they propose moves to neighboring states, where a natural neighborhood definition is the $k$-neighborhood

$$H_k(\boldsymbol{x}) := \{\boldsymbol{\gamma} \in \mathbb{B}^d \colon\ |\boldsymbol{x} - \boldsymbol{\gamma}| \leq k\}. \tag{1.9}$$

There is a variety of ways to propose new states from $H_k(\boldsymbol{x})$ and to choose the size of the neighborhood $k$. A standard auxiliary kernel is

$$q(\boldsymbol{\gamma} \mid \boldsymbol{x}) = \sum_{I \subseteq D} q(\boldsymbol{\gamma} \mid \boldsymbol{x}, I) \sum_{k \in D} \psi(I \mid k)\omega(k)$$

where $\omega$ is the distribution of the number $k$ of components to be changed in the proposal, $\psi(\cdot \mid k)$ is the uniform distribution on the set of all index sets $I$ with cardinality $k$, and $q(\cdot \mid \boldsymbol{x}, I)$ is a Bernoulli distribution with mean $\boldsymbol{m}_I$ for all components indexed by $I$ and a copy of $\boldsymbol{x}_{D \setminus I}$ for all other components. Explicitly the mass function is

$$q(\boldsymbol{\gamma} \mid \boldsymbol{x}) = \sum_{I \subseteq D} \prod_{i \in D \setminus I} \delta_{x_i}(\gamma_i) \prod_{i \in I} \left[ m_i(\boldsymbol{x})^{\gamma_i} [1 - m_i(\boldsymbol{x})]^{1-\gamma_i} \right] \sum_{k \in D} \frac{k!(d-k)!}{d!} \delta_k(|I|)\, \omega(k), \tag{1.10}$$

and sampling from $q(\cdot \mid \boldsymbol{x})$ is straightforward, see Procedure 1. In the following, we discuss some special cases.

**Procedure 1:** Generic random walk kernel

**Input**: $\boldsymbol{x} \in \mathbb{B}^d$

$u \sim \mathcal{U}_{[0,1]}$, $k \sim \omega$, $I \sim \psi(\cdot \mid k) = \mathcal{U}_{\{I \subseteq D \mid |I| = k\}}$

$\boldsymbol{y} \leftarrow \boldsymbol{x}$

**for** $i \in I$ **do** $y_i \sim m_i(\boldsymbol{x})^{y_i}[1 - m_i(\boldsymbol{x})]^{1-y_i}$

**if** $\dfrac{\pi(\boldsymbol{y})}{\pi(\boldsymbol{x})} \prod_{i \in I} \left[ \dfrac{m_i(\boldsymbol{x})}{1 - m_i(\boldsymbol{x})} \right]^{x_i - y_i} > u$ **then**

  **return** $\boldsymbol{y}$

**else**

  **return** $\boldsymbol{x}$

**end**

### Random scan Gibbs sampler

Suppose that $\omega = \delta_1$. Moves from $\boldsymbol{x}$ are restricted to $H_1(\boldsymbol{x})$ which is referred to as single site updating. The classic *random scan Gibbs* sampler draws an index $i \in D$ and samples the $i$th component from the full conditional distribution

$$\pi_i(\gamma_i \mid \boldsymbol{\gamma}_{-i}) = \frac{\pi(\boldsymbol{\gamma})}{\pi(\gamma_i = 1, \boldsymbol{\gamma}_{-i}) + \pi(\gamma_i = 0, \boldsymbol{\gamma}_{-i})}, \tag{1.11}$$

which corresponds to setting

$$m_i(\boldsymbol{x}) = \pi_i(1 \mid \boldsymbol{x}_{-i}), \quad i \in D.$$

Alternatively, a *deterministic scan* sampler would iterate through $\sigma(1), \ldots, \sigma(d)$ for a uniformly drawn permutation $\sigma \sim \mathcal{U}_{P_D}$ where $P_D := \{f \colon D \to D \mid f \text{ is bijective}\}$ which

may decrease the risk that the chain moves forth and back around the same local modes. Let $\tilde{\boldsymbol{x}}^{(i)}$ be a copy of $\boldsymbol{x}$ with $\tilde{x}_i^{(i)} = 1 - x_i$ for $i \sim \mathcal{U}_D$. By construction, the acceptance probability is

$$\alpha(\boldsymbol{x}, \tilde{\boldsymbol{x}}^{(i)}) = \frac{\pi(\tilde{\boldsymbol{x}}^{(i)})\pi(1 \mid \boldsymbol{x}_{-i})^{x_i - \tilde{x}_i^{(i)}}}{\pi(\boldsymbol{x})\pi(0 \mid \boldsymbol{x}_{-i})^{x_i - \tilde{x}_i^{(i)}}} = 1$$

while the average mutation probability is only

$$\mu(\boldsymbol{x}) = \frac{1}{d} \sum_{i \in D} \frac{\pi(\tilde{\boldsymbol{x}}^{(i)})}{\pi(\boldsymbol{x}) + \pi(\tilde{\boldsymbol{x}}^{(i)})}.$$

### Metropolized Gibbs sampler

Suppose that $\omega = \delta_1$. In comparison to the Gibbs sampler, we obtain a more efficient chain in terms of mutation rates (Liu, 1996b) using the simple form

$$m_i(\boldsymbol{x}) = 1 - x_i, \quad i \in D.$$

The scheme with deterministic flips is sometimes referred to as *metropolized Gibbs*, since one replaces the full conditional distribution by a Metropolis-Hasting type proposal. Since we always propose to change the current state, the acceptance probability becomes

$$\alpha(\boldsymbol{x}, \tilde{\boldsymbol{x}}^{(i)}) = \frac{\pi(\tilde{\boldsymbol{x}}^{(i)})}{\pi(\boldsymbol{x})} \wedge 1, \tag{1.12}$$

but the average mutation probability is

$$\mu(\boldsymbol{x}) = \frac{1}{d} \sum_{i \in D} \left[ \frac{\pi(\tilde{\boldsymbol{x}}^{(i)})}{\pi(\boldsymbol{x})} \wedge 1 \right]$$

and therefore higher than for the random Gibbs sampler. From the average mutation probabilities, we may conclude that a Markov chain with deterministic flips moves, on average, faster than the classical random scan Gibbs chain. This is particularly important if the mass function $\pi$ is expensive to compute.

### Uniform block updating

Suppose that $\omega \neq \delta_1$. Moves from $\boldsymbol{x}$ are not restricted to $H_1(\boldsymbol{x})$ which is often referred to as block updating, since one proposes to alter a block of entries in the Metropolis-Hastings step.

One maximizes the average mutation rate, conditional on the event that a move is accepted, by setting $m_i(\boldsymbol{x}) = 1 - x_i$ for all $i \in I \sim \psi(\cdot \mid k)$ and $k \sim \omega$. The auxiliary kernel simplifies

$$q(\boldsymbol{\gamma} \mid \boldsymbol{x}) = \sum_{k=1}^{d} \delta_k(|\boldsymbol{x} - \boldsymbol{\gamma}|) \frac{k!(d-k)!}{d!} \omega(k), \tag{1.13}$$

which is a generalization of the metropolized Gibbs kernel to block updating. The auxiliary kernel is symmetric in the sense that $q(\boldsymbol{\gamma} \mid \boldsymbol{x}) = q(\boldsymbol{x} \mid \boldsymbol{\gamma})$, and the Metropolis-Hastings ratio (1.7) simplifies to $[\pi(\boldsymbol{\gamma})/\pi(\boldsymbol{x})] \wedge 1$ where $\boldsymbol{\gamma} \sim q(\cdot \mid \boldsymbol{x})$ denotes the proposal.

#### Swendsen-Wang updating

Since the uniformly chosen update blocks do not take the distribution of interest into account, these blind moves are rarely accepted for large blocks. For binary distributions from the exponential multi-linear family (see Section 3.5.1 for details), the special structure of the mass function can be exploited to detect promising blocks.

Swendsen and Wang (1987) propose a sampling procedure that introduces a vector of auxiliary variables $\boldsymbol{u}$ such that $\pi(\boldsymbol{u} \mid \boldsymbol{\gamma})$ is a distribution of mutually independent uniforms and $\pi(\boldsymbol{\gamma} \mid \boldsymbol{u})$ a distribution with components which are either fixed by constraints or conditionally independent. Higdon (1998) suggests to parameterize and control the size of the conditionally independent blocks to further improve the mixing properties.

Nott and Green (2004) attempt to adapt the rationale behind the algorithm to sampling from a broader class of binary distributions. However, the Swendsen-Wang algorithm is based on the exponential multi-linear structure of the distribution of interest and the efficiency gain does not easily carry over to general binary sampling.

### 1.3.2. Metropolis-Hastings independence sampler

Suppose that $\omega = \delta_d$ and $m_i(\boldsymbol{x}) = m_i$ for all $i \in D$. The auxiliary kernel (1.10) becomes the product distribution

$$q_{\boldsymbol{m}}^{\sqcap}(\boldsymbol{\gamma}) = \textstyle\prod_{i=1}^{d} m_i^{\gamma_i}(1 - m_i)^{1-\gamma_i}, \tag{1.14}$$

and does not depend on the current state $\boldsymbol{x}$. The Metropolis-Hastings kernel with independent proposals is referred to as the *Metropolis-Hastings independence sampler*. The kernel $q(\cdot \mid \boldsymbol{x})$ simplifies to a distribution $q$ which needs to verify $\mathrm{supp}(\pi) \subseteq \mathrm{supp}(q)$ to ensure that the Markov chain is irreducible. The acceptance probability is

$$\alpha_q(\boldsymbol{x}, \boldsymbol{y}) = \frac{\pi(\boldsymbol{y})q(\boldsymbol{x})}{\pi(\boldsymbol{x})q(\boldsymbol{y})} \wedge 1. \tag{1.15}$$

Thus, the average acceptance rate and the average mutation rate

$$\mu_q(\boldsymbol{x}) = \sum_{\boldsymbol{\gamma}\in\mathbb{B}^d\setminus\{\boldsymbol{x}\}} \left[\frac{\pi(\boldsymbol{\gamma})q(\boldsymbol{x})}{\pi(\boldsymbol{x})q(\boldsymbol{\gamma})} \wedge 1\right] q(\boldsymbol{\gamma})$$

practically coincide on large sampling spaces. Obviously, in order to make this approach work, we need to choose $q$ sufficiently close to $\pi$. For the Metropolis-Hastings independence sampler, the average acceptance rate of the kernel

$$\overline{\alpha}_q := \sum_{\boldsymbol{x}\in\mathbb{B}^d}\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} \alpha_q(\boldsymbol{x},\boldsymbol{\gamma})\pi(\boldsymbol{x})q(\boldsymbol{\gamma}) \tag{1.16}$$

can be bounded from below by the total variation distance $1 - 2\|q - \pi\|_{tv}$. The second-largest eigenvalue of the transition kernel is

$$\lambda_2 = 1 - \min_{\boldsymbol{\gamma}\in\mathbb{B}^d}(q(\boldsymbol{\gamma})/\pi(\boldsymbol{\gamma})),$$

and the constant in (1.4) is $c(p) = [2\pi(\boldsymbol{x}_0)]^{-\frac{1}{2}}$ for $p = \delta_{\boldsymbol{x}_0}$, see Diaconis and Hanlon (1992) and Liu (1996a) for details on the eigenanalysis.

In most practical situations, the product proposal distribution (1.14) does not yield reasonable acceptance rates. Proposition 3.2.8 in Chapter 3 states that even if the distribution of interest $\pi$ and the auxiliary distribution $q$ both have the same mean $\boldsymbol{m} \in (0,1)^d$, the auto-correlation of the independent Metropolis-Hastings sampler heavily depends on the second cross-moments. In other words, if the distribution of interest features strong correlations between its components, the independent Metropolis-Hastings sampler using a vector of independent Bernoulli variables as proposal is bound to suffer from extremely low acceptance rates.

Therefore, to make a Metropolis-Hastings independence sampler work on $\mathbb{B}^d$ we have to provide a parametric family $(q_\theta)_{\theta\in\Theta}$ which is richer than (1.14) and we need to calibrate the parameter $\theta$ such that the distance between $q_\theta$ and $\pi$ is minimized. We come back to this Markov kernel as essential part of the SMC algorithm discussed in Chapter 2.

## 1.4. Adaptive Markov chain Monte Carlo

The Metropolis-Hastings sampler allows to incorporate any proposal kernel $q$ which satisfies $\text{supp}(\pi) \subseteq \text{supp}([\delta_{\boldsymbol{x}}q^n])$ for $n > n_0 \in \mathbb{N}$. But obviously not all choices yield good MCMC estimators. In most practical cases, one identifies a suitable family of auxiliary kernels $(q_\theta)_{\theta\in\Theta}$ but still faces the problem that the parameter $\theta$ needs to be calibrated against the distribution of interest $\pi$. The obvious idea is to improve the choice of $\theta$

during the course of the algorithm which is then referred to as *adaptive*. The transition kernels $(\kappa_\theta)_{\theta\in\Theta}$ all admit $\pi$ as invariant distribution but if we adapt the parameter $\theta_{n+1}$ in function of the sampled trajectory $(\boldsymbol{X}_k)_{k\in N_0} \sim \pi\kappa_{\theta_1}\cdots\kappa_{\theta_n}$, the chain becomes non-stationary and looses its Markov property. This raises the question whether the ergodic theorem still applies which justifies the MCMC estimator.

There has been a major interest in adaptive Markov chain Monte Carlo (AMCMC) in the recent years and convergence results have been established which hold on finite spaces under very mild conditions (Roberts and Rosenthal, 2007). For further details on AMCMC we refer to Andrieu and Thoms (2008) and citations therein. In the following, we review some AMCMC algorithms for sampling on binary spaces and propose a few extensions without going into details.

### 1.4.1. Adaptive metropolized Gibbs

An adaptive extension of the Gibbs sampler has been proposed by Nott and Kohn (2005). The authors also provide a direct proof of convergence for their AMCMC algorithms which needs less preparation than the rather technical proofs for general state spaces (Roberts and Rosenthal, 2007). The full conditional distribution is the optimal choice in terms of acceptance rates, but oftentimes the chain does not move because the current state has been sampled again; see the remark on the Gibbs sampler in Section 1.3.1. If the mass function of the distribution of interest is expensive to evaluate the Gibbs sampler is bound to waste a lot of computational time.

Nott and Kohn (2005) suggest to replace the expensive full conditional distribution $\pi(\gamma_j = 1 \mid \boldsymbol{\gamma}_{-j} = x_{-j})$ by a linear predictor. For the proposal kernel (1.10) let $\omega = \delta_1$ and

$$m_i(\boldsymbol{x}) := \left[\left(\psi_i - \frac{\mathbf{W}_{-i}\boldsymbol{x}_{-i}}{w_{i,i}}\right) \vee \delta\right] \wedge (1-\delta),$$

where $\psi$ is the estimated mean, $\mathbf{W}^{-1}$ the estimated covariance matrix and $\delta \in (0, 1/2)$ a design parameter which ensures that $p_i(\boldsymbol{x})$ is a probability. Analogously to our vector notation, $\mathbf{W}_{-i}$ denotes the matrix $\mathbf{W}$ without the $i$th row and column. The estimates are obtained from the past trajectory of the chain and updated periodically. The average mutation probability is of the same order as that of the Gibbs kernel, but adaption largely avoids computationally expensive evaluations of $\pi$: The non-adaptive Gibbs sampler already requires evaluation of $\pi$ to compute the sampling probability (1.11). In contrast, the adaptive metropolized Gibbs sampler only evaluates $\pi(\boldsymbol{\gamma})$ if $\boldsymbol{x} \neq \boldsymbol{\gamma}$ for the linear predictor proposal $\boldsymbol{\gamma} \sim q(\cdot \mid \boldsymbol{x})$.

### 1.4.2. Adaptive random walk

Lamnisos et al. (2011) propose to calibrate the distribution of the number of bits to be flipped on average, where they take $\omega = \mathcal{B}(\zeta, n)$ to be a binomial distribution with succes probability $\zeta$. Their work is motivated by the adaptive random walk algorithm developed by Atchadé and Rosenthal (2005) for continuous state spaces where the variance of the multivariate normal random walk proposal is adjusted to meet the (asymptotically) optimal acceptance probability. However, in the context of binary spaces the major problem practitioners are facing is multi-modality, see Section 1.5. The method proposed by Atchadé and Rosenthal is designed for high-dimensional unimodal sampling problems, and the rationale behind the design of the algorithm does therefore not necessarily carry over to multi-modal discrete problems.

Deville and Tillé (2004) propose a method developed in the context of survey sampling as a variance reduction technique for the Horvitz–Thompson estimator referred to as the *cube method*, which allows to sample from the product family $q^{\square}_{\boldsymbol{m}}$ defined in (1.14) conditional on a set of linear constraints. Their algorithm yields an alternative random walk scheme which has, to our knowledge, not been proposed in the context of AMCMC on binary spaces. Instead of a random walk on the neighborhood (1.9), one would perform a random walk on

$$K_a(\boldsymbol{x}) = \{\boldsymbol{\gamma} \in \mathbb{B}^d \colon\ |\boldsymbol{x}| - a \leq |\boldsymbol{\gamma}| \leq |\boldsymbol{x}| + a\},$$

that is the neighborhood of models with a number predictors differing by less than $a$. Given the current state $\boldsymbol{x}$, we first draw the number of predictors $k$ uniformly from the set $[\![0 \vee (|\boldsymbol{x}| - a), d \wedge (|\boldsymbol{x}| - a)]\!]$. The proposal $\boldsymbol{\gamma}$ is drawn from $q^{\square}_{\boldsymbol{m}}$ conditional on the event that $|\boldsymbol{\gamma}| = k$, where the mean $\boldsymbol{m}$ needs to be adapted during the run of the MCMC.

The conditional sampling problem is not trivial, since the mean of each component $m_i$ may be different. The name *cube method* stems from the idea to construct a vector $\boldsymbol{v} \in \mathbb{R}^d$ in the kernel of the linear constraints and determine the two facets of the hyper-cube $[0, 1]^d$ it intersects. A random draw, with probabilities proportional to the distance between the starting point $\boldsymbol{m}$ and the facets, determines one facet to be fixed, and the iteration is repeated on the remaining hyper-cube $[0, 1]^{d-1}$ until all facets are fixed. The construction of the vectors may be deterministic which also allows to evaluate the mass function which is necessary in the Metropolis-Hastings step. We refer to Deville and Tillé (2004) for details on this technique.

### 1.4.3. Adaptive independence sampler

The Metropolis-Hastings independence sampler is rapidly mixing if we can fit the auxiliary distribution $(q_\theta)_{\theta\in\Theta}$ to be sufficiently close to the target distribution $\pi$. Unfortunately, we face a hen-and-egg problem since the non-adaptive Markov chain is likely to mix very poorly but without any significant state space exploration we cannot reasonably adapt $(\kappa_\theta)_{\theta\in\Theta}$. A viable solution is to mix the Metropolis-Hastings independence kernel $\kappa_\theta$ and a non-adaptive random walk kernel $\kappa_{\mathrm{RW}}$

$$\kappa_\varrho = (1-\varrho)\kappa_{\mathrm{RW}} + \varrho\kappa_\theta$$

for some parameter $\varrho \in [0,1]$. The sampler proposes an independently drawn state with probability $\varrho$, which may be increased adaptively during the run of the MCMC after the parameter of the proposal distribution $\theta$ has been adapted sufficiently.

## 1.5. Multi-modality

We briefly motivate why the MCMC methods discussed in Section 1.2 might fail to provide reliable estimates of the expected value (1.1) if the distribution of interest $\pi$ is strongly multi-modal. There does not seem to be a precise mathematical definition of multi-modality since this notion is somewhat diffuse.

We say that $\boldsymbol{x} \in \mathbb{B}^d$ is a local mode of degree $k$ if $\pi(\boldsymbol{x}) \geq \pi(\boldsymbol{\gamma})$ for all $\boldsymbol{\gamma} \in H_k(\boldsymbol{x})$. We call $\pi$ a strongly multi-modal distribution if there is a significant collection $M$ of local modes of moderate degrees and mass function values $\pi(\boldsymbol{x}) \gg 2^{-d}$ for all $\boldsymbol{x} \in M$. These distributions are difficult to sample from using random walk MCMC methodology since we have to ensure that the trajectory of the Markov chain covers all regions of interest in order to appeal to the ergodic theorem.

### 1.5.1. Markov chains and multi-modality

Transition kernels of the symmetric type are known to be slowly mixing on multi-modal problems. If we put most weight on small values of $k$, the Markov chain is bound to remain in the region of a single local mode for a long time. If we put more weight on larger values of $k$, the proposals will hardly ever be accepted unless we propose by chance a state in the domain of another local mode. Obviously, there is a problem dependent trade-off when choosing the distribution $\omega$.

Adaptive MCMC algorithms provide an astonishing speed-up over their non-adaptive versions for high-dimensional sampling problems on continuous spaces and unimodal distributions of interest. Still, it is a notoriously difficult problem to adapt an MCMC sampler to a multi-modal sampling problem. Premature adaption might even worsen the estimator by providing the impression of good mixing on just a subset of the state space. There are more advanced MCMC algorithms which use parallel tempering ideas combined with more elaborate local moves (Bottolo and Richardson, 2010, among others) or self-avoiding dynamics (Hamze et al., 2011) to overcome the multi-modality problem. However, these algorithms seem difficult to tune automatically.

### 1.5.2. Bayesian adaptive sampling

As an alternative to MCMC sampling, Clyde et al. (2011) develop the Bayesian adaptive sampling procedure which draws binary vectors without replacement and uses the normalized estimator (1.6). The idea is to update the conditional probabilities to ensure that each binary vector is only sampled once. The algorithm starts sampling with some initial mean $\boldsymbol{m}_0$ which is then updated using current estimate $\hat{\boldsymbol{m}}_n$ of the mean of interest. The updating of the conditional probabilities is rather expansive and has to be compromised in practice, meaning that the updating step cannot be performed after every single sampling step. From a computational perspective this seems reasonable.

However, the critical problem is that the method does not sample from the distribution of interest but from a sequence of distributions $q^{\square}_{\hat{\boldsymbol{m}}_n}$ with a mean $\hat{\boldsymbol{m}}_n$ that needs to be estimated during the course of the algorithm. The authors' claim that this sequence is "close" to the target distribution is disputable. Even if the mean was correct, an IS estimator of $\pi$ based on proposals drawn from $q^{\square}_{\boldsymbol{m}^\pi}$ might be quite inefficient in the presence of strong multi-modality.

The rationale to produce a unique collection $V$ of the most likely models leads to stochastic search methods which identify a collection of local modes which may be averaged according to their posterior mass. This has been proposed for inference in state spaces which are clearly too large to achieve approximate ergodicity with standard MCMC methods, see e.g. Hans et al. (2007). We discuss optimization algorithms on binary spaces in Chapter 6.

# 2. The sequential Monte Carlo sampler

## Resumé

*Nous proposons une technique d'échantillonnage alternative basée sur les méthodes de Monte-Carlo séquentielles qui propage un système de particules à partir d'une loi initiale simple, par des lois intermédiaires auxiliaires vers la loi cible. Alors que la méthodologie resample-move provient de la boîte à outils standard du filtrage particulaire (Del Moral et al., 2006), l'innovation centrale est l'utilisation d'un noyau de Metropolis-Hastings avec des propositions indépendantes dans l'étape de déplacement. L'usage des familles paramétriques avancées qui approchent efficacement les lois intermédiaires et permettent d'atteindre des taux d'acceptation élevés nécessaires pour la construction de chaînes de Markov rapidement mélangeantes.*

## 2.1. Introduction

In this chapter, we introduce a fully adaptive resample-move algorithm for sampling from binary distribution using sequential Monte Carlo (SMC) methodology. The material has been published in Schäfer and Chopin (2012) and partially extended in Schäfer (2012b). We discuss how to obtain estimates of expected values of the form (1.1) providing a self-contained description of the SMC framework. In particular, we propose some novel ideas tailored to sampling on binary spaces. For a more general overview of SMC methods we refer to Del Moral et al. (2006).

The basic resample-move algorithm alternates importance sampling steps, resampling steps and Markov chain transitions, to recursively approximate a sequence of distributions $(\pi_t)_{t\in\mathbb{N}}$, using a set of weighted 'particles' $(\boldsymbol{w}_t, \mathbf{X}_t)$ which provide an empirical representation of the current distribution. This sequence of distributions is chosen to finally provide a particle system which approximates the distribution of interest $\pi = \pi_\tau$ and thus yield an estimator

$$\hat{\pi}^n_{\text{SMC}}(f) = \sum_{k=1}^n w_{k,\tau} f(\boldsymbol{X}_{k,\tau}), \tag{2.1}$$

where $n$ is the number of particles. Under mild conditions, Chopin (2004) shows that the estimator is consistent and asymptotically normal, in particular $\hat{\pi}^n_{\text{SMC}}(f) \overset{n\to\infty}{\longrightarrow} \pi(f)$ a.s. The details of the SMC sampler summarized in Algorithm 2 are discussed in separate steps in the upcoming sections.

**Algorithm 2:** Resample-move

**Input**: $f\colon \mathbb{B}^d \to \mathbb{R}$
**for all** $k \in N$ **sample** $\boldsymbol{x}_k \sim p$.
**while do**
  $\alpha \leftarrow$ **find step length**$(\varrho, \mathbf{X})$ (Procedure 4)
  $\boldsymbol{w} \leftarrow$ **importance weights**$(\alpha, \pi_\varrho, \mathbf{X})$ (Procedure 3)
  $\varrho \leftarrow \varrho + \alpha$
  **if** $\varrho \equiv 1$ **then return** $\sum_{k=1}^n w_k f(\boldsymbol{x}_k)$
  $\theta \leftarrow$ **fit parametric family**$(\boldsymbol{w}, \mathbf{X})$ (see Chapter 3)
  $\widehat{\mathbf{X}} \leftarrow$ **resample**$(\boldsymbol{w}, \mathbf{X})$ (Procedure 5)
  $\mathbf{X} \leftarrow$ **move**$(\kappa_\theta, \widehat{\mathbf{X}})$ (Procedure 6)
**end**

## 2.2. Sequential Importance Sampling

The first ingredient of the SMC sampler is a sequence of distributions $(\pi_t)_{t\in\mathbb{N}}$ that serves as a bridge between some easy initial distribution and the distribution of interest. The intermediary distributions $\pi_t$ are purely instrumental. The idea is to depart from a distribution $p$ with broad support and to progress smoothly towards $\pi$.

We construct a smooth sequence of distributions by judicious choice of an associated real sequence $(\varrho_t)_{t=0}^\tau$ increasing from zero to one. The most convenient and somewhat natural strategy is a sequence of elements from the geometric bridge (Gelman and Meng, 1998; Neal, 2001; Del Moral et al., 2006)

$$\pi_\varrho \propto p^{1-\varrho}\pi^\varrho, \quad \varrho \in [0,1]. \tag{2.2}$$

One could also take a sequences of from a family of mixtures $\pi^{(m)}_\varrho \propto (1-\varrho)p + \varrho\pi$ but this is computationally less convenient. We discuss some alternative choices for sequences in the context of particular applications in Chapters 4, 5 and 6. The question how to actually choose an appropriate sequence $(\pi_{\varrho_t})_{t\in\mathbb{N}}$ from $(\pi_\varrho)_{\varrho\in[0,1]}$ is addressed in the next section.

### 2.2.1. Importance weights

Following standard sequential Monte Carlo notation, we refer to

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

with $|\boldsymbol{w}| = 1$ as a particle system with $n$ particles. We say the particle system $(\boldsymbol{w}, \mathbf{X})$ *targets* a probability distribution $q$ if the empirical distribution converges

$$\textstyle\sum_{k=1}^n w_k \, \delta_{\boldsymbol{x}_k} \overset{n\to\infty}{\longrightarrow} q, \quad \text{a.s.}$$

Suppose we have produced a sample $\boldsymbol{x}_{1,t}, \dots, \boldsymbol{x}_{n,t}$ of size $n$ from $\pi_t$. We can roughly approximate $\pi_{t+1}$ by the empirical distribution

$$\pi_{t+1}(\boldsymbol{\gamma}) \approx \textstyle\sum_{k=1}^n w_{t+1}(\boldsymbol{x}_{k,t}) \, \delta_{\boldsymbol{x}_{k,t}}(\boldsymbol{\gamma}), \tag{2.3}$$

where the corresponding importance function $w_{t+1}$ is

$$w_{t+1}(\boldsymbol{x}) := \frac{u_{t+1}(\boldsymbol{x})}{\sum_{k=1}^n u_{t+1}(\boldsymbol{x}_{k,t})}, \quad u_{t+1}(\boldsymbol{x}) := \frac{\pi_{t+1}(\boldsymbol{x})}{\pi_t(\boldsymbol{x})}. \tag{2.4}$$

As we choose $\pi_t$ further from $\pi_{t-1}$, the weights become more uneven and the accuracy of the importance approximation deteriorates. If we repeat the weighting steps until we reach $\pi$, we obtain a classical importance sampling estimate with instrumental distribution $p$ which is in most cases a very poor estimator. The idea of the SMC algorithm is to monitor the effective sample size (ESS) estimate $\hat{\eta}_n$ defined in (1.2) and intersperse resample and move steps before loosing track of the particle approximation.

**Procedure 3:** Importance weights

**Input**: $\alpha$, $\pi$, $\mathbf{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)^\intercal$

$u_k \leftarrow \pi^\alpha(\boldsymbol{x}_k)$ **for all** $k \in N$

$w_k \leftarrow u_k / (\sum_{i=1}^n u_i)$ **for all** $k \in N$

**return** $\boldsymbol{w} = (w_1, \dots, w_n)$

### 2.2.2. Optimal step length

Given any sequence $(\pi_t)_{t\in\mathbb{N}}$ bridging the gap between $p$ to $\pi$, we could repeatedly reweight the system and monitor whether the ESS falls below some critical threshold like one does in particle filtering applications like target tracking. However, in the static context the sequence $(\pi_t)_{t\in\mathbb{N}} = (\pi_{\varrho_t})_{t\in\mathbb{N}}$ comes from a family $(\pi_\varrho)_{\varrho\in[0,1]}$, and one may exactly control the weight degeneracy by judicious choice of the step lengths $\alpha_t = \varrho_{t+1} - \varrho_t$.

The ESS after weighting $\hat{\eta}_n(\boldsymbol{w}_{t,\alpha})$ is merely a function of $\alpha$. For an unweighted particle system $\mathbf{X}_t$ at time $t$, we pick a step length such that

$$\hat{\eta}_n(\boldsymbol{w}_{t,\alpha}) = \eta^*, \tag{2.5}$$

that is we lower the ESS with respect to the current particle approximation by some fixed ratio $\eta^* \in (0, 1)$ (Jasra et al., 2011; Del Moral et al., 2012). This ensures a 'smooth' transition between two auxiliary distributions, in the sense that consecutive distributions are close enough to approximate each other reasonably well using importance weights.

We obtain the associated sequence $(\varrho_t)_{t\in\mathbb{N}}$ by setting $\varrho_{t+1} = \varrho_t + \alpha_t$ where $\alpha_t$ is a unique solution of (2.5) which is easily obtained using bi-sectional search since $\hat{\eta}_n(\boldsymbol{w}_{t,\alpha})$ is continuous and monotonously decreasing in $\alpha$, see Procedure 4. This is particularly fast to compute for the geometric bridge since $u_t(\boldsymbol{x}) = [\pi(\boldsymbol{x})/p(\boldsymbol{x})]^{\alpha_t}$.

For fixed $\eta^*$, the associated sequence $(\varrho_t)_{t\in\mathbb{N}}$ is a self-tuning parameter but the number of steps until termination of the SMC algorithm is not known in advance and largely depends on the speed parameter $\eta^*$ and the complexity of the sampling problem at hand. In our simulations, we choose $\eta^* = 0.92$ yielding good results on all example problems of moderate dimension $d \sim 100$. As the dimension of the sampling problem increases, we have to progress more slowly and thus choose $\eta^*$ closer to one.

**Procedure 4:** Find step length

**Input**: $\varrho$, $\mathbf{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)^\intercal$
$l \leftarrow 0$, $u \leftarrow 1.05 - \rho$, $\alpha \leftarrow 0.05$
**repeat**
  **if** $\eta(\alpha, \mathbf{X}) < \eta^*$ **then** $u \leftarrow \alpha$, $\alpha \leftarrow (\alpha + l)/2$
  **else** $l \leftarrow \alpha$, $\alpha \leftarrow (\alpha + u)/2$
**until** $|u - l| < \varepsilon$ **or** $l > 1 - \varrho$;
**return** $\alpha \wedge (1 - \varrho)$

### 2.2.3. Resampling step

We replace the system $(\boldsymbol{w}_{t+1}, \mathbf{X}_t)$ targeting $\pi_{t+1}$ by a selection of particles $\hat{\boldsymbol{x}}_1, \dots, \hat{\boldsymbol{x}}_n$ drawn from the current particle reservoir $\boldsymbol{x}_{1,t}, \dots, \boldsymbol{x}_{n,t}$ such that

$$\mathbb{E}\left(n(\boldsymbol{x}_k)\right) = n\, w_k,$$

where $n(\boldsymbol{x})$ denotes the number of particles identical with $\boldsymbol{x}$. Thus, in the resampled system, particles with small weights have vanished while particles with large weights

have been multiplied. For the implementation of the resampling step, there exist several recipes. We could apply a multinomial resampling (Gordon et al., 1993) which is straightforward. There are, however, more efficient ways like residual (Liu and Chen, 1998), stratified (Kitagawa, 1996) and systematic resampling (Carpenter et al., 1999) which are variance reduction techniques that improve the SMC estimator. We refer to Douc et al. (2005) for a detailed comparison. In our simulations, we always used the systematic resampling scheme, see Procedure 5.

**Procedure 5:** Systematic resampling step

**Input:** $\boldsymbol{w} = (w_1, \ldots, w_n)$, $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\intercal$
$\boldsymbol{v} \leftarrow n\,\boldsymbol{w}$, $i \leftarrow 1$, $c \leftarrow v_1$
**sample** $u \sim \mathcal{U}_{[0,1]}$
**for** $k = 1$ **to** $n$ **do**
  **while** $c < u$ **do** $i \leftarrow i + 1$, $c \leftarrow c + v_i$
  $\hat{\boldsymbol{x}}_k \leftarrow \boldsymbol{x}_i$, $u \leftarrow u + 1$
**end**
**return** $\widehat{\mathbf{X}} = (\hat{\boldsymbol{x}}_1 \ldots, \hat{\boldsymbol{x}}_n)^\intercal$

## 2.3. Adaptive move step

### 2.3.1. Fast-mixing kernels

The resampling step provides an unweighted particle system of $\pi_t$ containing multiple copies of many particles. The central idea of the SMC algorithm is to diversify the resampled system by draws from a Markov kernel which admits the current target distribution as invariant measure (Gilks and Berzuini, 2001). The particle $\hat{\boldsymbol{x}}_{k,t+1}^{(0)}$ is approximately distributed according to $\pi_{t+1}$, and a draw

$$\hat{\boldsymbol{x}}_{k,t+1}^{(1)} \sim \kappa_{t+1}(\cdot \mid \hat{\boldsymbol{x}}_{k,t+1}^{(0)})$$

from a kernel with $[\pi_{t+1}\kappa_{t+1}] = \pi_{t+1}$ is again approximately distributed according to $\pi_{t+1}$. The last sample of the generated Markov chain $(\hat{\boldsymbol{x}}_{k,t+1}^{(0)}, \ldots, \hat{\boldsymbol{x}}_{k,t+1}^{(s)})$ is, for sufficiently many move steps $s \in \mathbb{N}$, almost exactly distributed according to the invariant measure $\pi_{t+1}$ and independent of its starting point.

In order to make the algorithm practical,the transition kernel needs to be rapidly mixing and diversify the particle system within just a few steps. The novel idea is to use a Metropolis-Hastings independence sampler as described in Section 1.3.2. The proposal

distribution is a parametric family $(q_\theta)_{\theta\in\Theta}$ which is, for a well-chosen parameter $\hat{\theta}_{t+1}$s, sufficiently close to $\pi_{t+1}$ to allow for reasonable acceptance probabilities. The parameter $\hat{\theta}_{t+1}$ is estimated based on the current particle approximation $(\boldsymbol{w}_{t+1}, \mathbf{X}_t)$ of $\pi_{t+1}$, as proposed in Chopin (2002). The choice of the parametric family is crucial and further discussed in Chapter 3. The locally operating Markov kernels reviewed in Section 1.2 are less suitable for the SMC algorithm since they mix rather slowly. However, batches of local moves can be alternated with independent proposals to ensure that the algorithm explores the neighborhood of local modes sufficiently well.

### 2.3.2. Adaptive stopping rule

While we could always apply a fixed number $s \in \mathbb{N}$ of move steps, we rather use an adaptive stopping criterion based on the number of distinct particles. We define the particle diversity as

$$\zeta_n(\mathbf{X}) := n^{-1}|\{\boldsymbol{x}_k \colon k \in N\}|. \tag{2.6}$$

Ideally, the sample diversity $\zeta_n(\mathbf{X})$ should correspond to the expected diversity

$$\zeta_n(\pi) := 1 \wedge n^{-1} \textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} \mathbb{1}_{\{\boldsymbol{x}\in\mathbb{B}^d \colon c_n\pi(\boldsymbol{x})\geq 1\}}(\boldsymbol{\gamma}),$$

where $c_n$ is the smallest value that solves $\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} \lfloor c_n\pi(\boldsymbol{\gamma})\rfloor \geq n$. This is the particle diversity we would expect if we had an independent sample from $\pi$. Therefore, if $\kappa_{t+1}$ is fast-mixing, we want to move the system until

$$\zeta_n(\widehat{\mathbf{X}}_{t+1}^{(s)}) \approx \zeta_n(\pi_{t+1}).$$

Since the quantity on the right hand side is unknown, we stop moving the system as soon as the particle diversity reaches a steady state we cannot push it beyond.

More precisely, we stop if the absolute diversity is above a certain threshold $\zeta^* \approx 0.95$ or the last improvement of the diversity is below a certain threshold $\zeta^*_\Delta > 0$. We always stop after a finite number of steps but the thresholds $\zeta^*$ and $\zeta^*_\Delta$ need to be calibrated to the efficiency of the transition kernel. For slow-mixing kernels, we recommend to perform batches of consecutive move steps instead of single move steps.

If the average acceptance rate $\overline{\alpha}$ of the kernel as defined in (1.16) is smaller than $\zeta^*_\Delta$, it is likely that the algorithm stops after the first iteration although further moves would have been necessary. We could adaptively adjust the threshold $\zeta^*_\Delta$ to be proportional to an estimate of the average acceptance rate; for our numerical experiments, however, we kept it fixed to $\zeta^*_\Delta \approx 10^{-2}$.

**Procedure 6:** Adaptive move step

**Input**: $\mathbf{X}^{[0]} = (\boldsymbol{x}_1^{[0]}, \dots, \boldsymbol{x}_n^{[0]}) \sim \hat{\pi}_t$, $\kappa_t$ **with** $[\pi_t \kappa] = \pi_t$
$s \leftarrow 1$
**repeat**
  **for all** $k \in N$ **sample** $\boldsymbol{x}_k^{(s)} \sim \kappa(\cdot \mid \boldsymbol{x}_k^{(s-1)})$
**until** $\zeta(\mathbf{X}^{(s)}) - \zeta(\mathbf{X}^{(s-1)}) < \zeta_\Delta^*$ **or** $\zeta(\mathbf{X}^{(s)}) > \zeta^*$
**return** $\mathbf{X}^{(s)} = (\boldsymbol{x}_1^{(s)} \dots, \boldsymbol{x}_n^{(s)})^\intercal$

## 2.4. Remark on discrete state spaces

Since the sample space $\mathbb{B}^d$ is discrete, a given particle is not necessarily unique. This raises the question whether it is sensible to store multiple copies of the same weighted particle in the system. Let

$$n(\boldsymbol{\gamma}) := \sum_{k \in N} \delta_{\boldsymbol{x}_k}(\boldsymbol{\gamma})$$

denote the number of copies of the particle $\boldsymbol{\gamma}$ in the system $(\boldsymbol{w}, \mathbf{X})$. Indeed, for parsimonious reasons, we could just keep a single representative of $\boldsymbol{\gamma}$ and aggregate the associated weights to $\tilde{w}(\boldsymbol{\gamma}) = n(\boldsymbol{\gamma})\, w(\boldsymbol{\gamma})$.

### 2.4.1. Impact on the effective sample size

Shifting weights between identical particles does not affect the nature of the particle approximation but it obviously changes the effective sample size $\eta_n(\boldsymbol{w})$ which is undesirable since we introduced the ESS as a criterion to measure the goodness of a particle approximation. From an aggregated particle system, we cannot distinguish the weight disparity induced by reweighting according to the importance function (2.4) and the weight disparity induced by multiple sampling of the same states which occurs if the mass of the target distribution is concentrated. More precisely, we cannot tell whether the ESS is actually due to the gap between $\pi_t$ and $\pi_{t+1}$ or due to the presence of particle copies as the mass of $\pi_t$ concentrates which occurs by construction of the auxiliary distribution in Section 6.1.1.

### 2.4.2. Impact on the resample-move step

Aggregating the weights means that the number of particles is not fixed at runtime. In this case, the straightforward way to implement the move step presented in Section 2.3.1 is breaking up the particles into multiple copies corresponding to their weights and

moving them separately. But instead of permanently splitting and pooling the weights it seems more efficient to just keep the multiple copies.

We could, however, design a different kind of resample-move algorithm which first augments the number of particles in the move step and then resamples exactly $n$ weighted particles from this extended system using a variant of the resampling procedure proposed by Fearnhead and Clifford (2003). A simple way to augment the number of particles is sampling and reweighting via

$$\boldsymbol{x}_k^{(1)} \sim q_{t+1}(\cdot \mid \boldsymbol{x}_k^{(0)}), \quad w_k^{(1)} = w_k \alpha, \; w_k^{(0)} = w_k(1-\alpha),$$

where $\alpha = \alpha_{q_{t+1}}(\boldsymbol{x}_k^{(1)}, \boldsymbol{x}_k^{(0)})$ denotes the acceptance probability (1.7) of the Metropolis-Hastings kernel. We tested this variant but could not see any advantage over the standard sampler presented in the preceding sections. For the augment-resample type algorithm the implementation is more involved and the computational burden significantly higher. In particular, the Rao-Blackwellization effect one might achieve when replacing the accept-reject steps of the transition kernel by a single resampling step does not seem to justify the extra computational effort.

Indeed, aggregating the weights does not only prevent us from using the ESS criterion, but also requires extra computational time of $\mathcal{O}(n \log n)$ in each iteration of the move step since pooling the weights is as complex as sorting. In the context of estimating an expected value, however, computational time is more critical than memory, and we therefore recommend to refrain from aggregating the weights.

# 3. Parametric families on binary spaces

## Resumé

*La performance de l'échantillonneur de Monte Carlo séquentiel dépend de la capacité d'échantillonner selon des lois auxiliaires qui sont, en un certain sens, proche à la loi de l'intérêt. Ce chapitre contient le travail principal de cette thèse et présente des stratégies visant à construire des familles paramétriques pour l'échantillonnage de vecteurs binaires avec dépendances. Nous proposons des solutions pratiques qui peuvent être incorporées dans les algorithmes particulaires sur les espaces binaires, mais aussi des approches de modélisation de vecteurs binaires aléatoires qui sont au-delà de l'application immédiate de méthodes Monte-Carlo. L'intérêt pratique des familles paramétriques proposées est examiné dans une étude numérique sur des matrices aléatoires de moments croisés.*

## 3.1. Motivation

The preceding chapters motivated why parametric families are an important building block of adaptive Monte Carlo algorithms. In this chapter, we elaborate on strategies for constructing parametric families which are suitable sampling distributions within and beyond the context of the sequential Monte Carlo sampler. Two major approaches to constructing parametric families are presented, based on generalized linear models or on multivariate copulas. We also review additive and multiplicative interactions which are not suitable for general purpose Monte Carlo algorithms but give insight in structural problems we face when designing parametric families. Finally, numerical experiments were performed to compare competing approaches for sampling binary data with specified mean and correlations in moderately high dimensions.

In the sequel, we summarize and discuss the conditions a parametric family $q$ with $\text{supp}(q) = \mathbb{B}^d$ should meet for successful integration into adaptive Monte Carlo algorithms, pointing out three approaches of practical value. This material is mostly taken from Schäfer (2012a).

(a) For reasons of parsimony, we prefer a family of distributions with at most $d(d+1)/2$ parameters like the multivariate normal on continuous spaces.

(b) Given a sample $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ from the distribution of interest $\pi$, one needs to compute an estimate $\hat{\theta}$ under the model $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \sim q_\theta^n$ within a reasonable amount of computational time.

(c) The family $q_{\theta \in \Theta}$ must allow to efficiently generate independent samples.

(d) In the context of an sequential Monte Carlo (SMC) or Markov chain Monte Carlo (MCMC) algorithm, the mass function $q_{\theta \in \Theta}(\cdot)$ needs to be evaluated point-wise. Note, however, that the cross-entropy (CE) method reviewed in Chapter 6 works without this requirement.

(e) The family $q_{\theta \in \Theta}$ needs to be sufficiently flexible to reproduce important characteristics of $\pi$, for example the mean and correlation structure, to ensure that the calibrated family $q_{\hat{\theta}}$ is sufficiently close to $\pi$.

The ultimate goal is to construct parametric families with $d(d+1)/2$ parameters which, like the multivariate normal, accommodate the full range of means and correlations on high-dimensional binary spaces. In the following, we provide an overview of three parametric families which seem useful in the context of adaptive Monte Carlo and comment on the requirement list composed above.

### 3.1.1. Product family

The simplest non-trivial distributions on $\mathbb{B}^d$ are certainly those having independent components. For a vector $\mathbf{m} \in (0,1)^d$ of marginal probabilities, consider the product family

$$q_{\boldsymbol{m}}^{\square}(\boldsymbol{\gamma}) := \textstyle\prod_{i=1}^d m_i^{\gamma_i}(1-m_i)^{1-\gamma_i}.$$

The product family meets most of the requirements. (a) The product family is parsimonious with $\dim(\theta) = d$. (b) The maximum likelihood estimator $\hat{\boldsymbol{m}}$ is the sample mean. (c) We can sample $\boldsymbol{y} \sim q_{\boldsymbol{m}}^{\square}$ by construction. (d) We can evaluate the mass function $q_{\boldsymbol{m}}^{\square}(\boldsymbol{y})$ by construction. (e) However, the product family does not reproduce any dependencies we might observe in the data $\mathbf{X}$.

The last point is the crucial weakness which makes the product family impractical for particle algorithms on strongly multi-modal problems. For toy examples which demonstrate this effect we refer to the applications in Sections 4.4.2 and 6.4.1.

### 3.1.2. Logistic conditionals family

For a lower triangular matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, consider the logistic conditionals family

$$q^{\ell}_{\mathbf{A}}(\boldsymbol{\gamma}) := \prod_{i=1}^{d} \ell\left(a_{ii} + \sum_{j=1}^{i-1} a_{ij}\gamma_j\right)^{\gamma_i} \left[1 - \ell\left(a_{ii} + \sum_{j=1}^{i-1} a_{ij}\gamma_j\right)\right]^{1-\gamma_i}$$

where $\ell\colon \mathbb{R} \to (0,1),\ \ell(x) = [1+\exp(-x)]^{-1}$ is the logistic function. The first component $\gamma_1$ is an independent Bernoulli variable; the $i$th component $\gamma_i$ conditional on $\boldsymbol{\gamma}_{1:i-1}$ is a logistic regression on the predictors $\gamma_1 \ldots, \gamma_{i-1}$.

The logistic conditionals family meets all of the requirements. (a) The logistic conditionals family is sufficiently parsimonious with $\dim(\theta) = d(d+1)/2$. (b) We can fit the parameter $\mathbf{A}$ via likelihood maximization. The fitting is computationally intensive but feasible. (c) We can sample $\boldsymbol{y} \sim q^{\ell}_{\mathbf{A}}$ by construction. (d) We can exactly evaluate $q^{\ell}_{\mathbf{A}}(\boldsymbol{y})$ by construction. (e) The family $q^{\ell}_{\mathbf{A}}$ reproduces the dependency structure of the data $\mathbf{X}$ although we cannot explicitly compute the marginal probabilities. The family is sufficiently flexible to reproduce any feasible combination of marginals and correlation structure.

### 3.1.3. Gaussian copula family

For a vector $\boldsymbol{a} \in \mathbb{R}^d$ and a correlation matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$, we introduce the mapping

$$\tau_{\boldsymbol{a}}\colon \mathbb{R}^d \to \mathbb{B}^d,\ \tau_{\boldsymbol{a}}(\boldsymbol{v}) := (\mathbb{1}_{(-\infty,a_i]}(v_1), \ldots, \mathbb{1}_{(-\infty,a_d]}(v_d)),$$

and consider the Gaussian copula family

$$q^{n}_{\boldsymbol{a},\boldsymbol{\Sigma}}(\boldsymbol{\gamma}) := (2\pi)^{-\frac{d}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \int_{\tau_{\boldsymbol{a}}^{-1}(\boldsymbol{\gamma})} \exp\left(-\tfrac{1}{2}\, \boldsymbol{v}^{\intercal}\boldsymbol{\Sigma}^{-1}\boldsymbol{v}\right)\, d\boldsymbol{v}.$$

The Gaussian copula family meets most of the requirements. (a) The Gaussian copula family is sufficiently parsimonious with $\dim(\theta) = d(d+1)/2$. (b) We can fit the parameters $\boldsymbol{a}$ and $\boldsymbol{\Sigma}$ via method of moments. However, the parameter $\boldsymbol{\Sigma}$ is not always positive definite. (c) We can sample $\boldsymbol{y} \sim q^{n}_{\boldsymbol{a},\boldsymbol{\Sigma}}$ using $\boldsymbol{y} = \tau_{\boldsymbol{a}}(\boldsymbol{v})$ with $\boldsymbol{v} \sim \varphi_{\boldsymbol{\Sigma}}$. (d) We cannot easily evaluate $q^{n}_{\boldsymbol{a},\boldsymbol{\Sigma}}(\boldsymbol{y})$ since this requires computing high-dimensional integral expressions which is a computationally challenging problem in itself (see e.g. Genz and Bretz (2009)). The Gaussian copula family is therefore less useful for SMC samplers but can be incorporated into the CE method analyzed in Chapter 6. (e) The family $q^{n}_{\boldsymbol{a},\boldsymbol{\Sigma}}$ reproduces the exact mean and, possibly scaled, correlation structure.

## 3.2. Preliminaries on random binary vectors

In the sequel, we elaborate some theoretical background on random binary vectors and provide a summary of known and novel results on modeling binary data with dependencies. Most of the material has been published as technical report (Schäfer, 2012a) which is under review for publication at the time this thesis is written.

### 3.2.1. Cross-moments and correlations

Before we discuss how to model dependencies in binary data, we introduce the notion of cross-moments and derive some elementary properties.

**Definition 3.2.1.** For a set $I \subseteq D$, we refer to

$$m_I^\pi := \mathbb{E}_\pi \left( \prod_{i \in I} X_i \right) = \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} \pi(\boldsymbol{\gamma}) \prod_{i \in I} \gamma_i$$

as the (absolute) cross-moment indexed by $I$.

Note that $m_I^\pi = \mathbb{P}_\pi \left( \boldsymbol{X}_I = \mathbf{1} \right)$ which means that cross-moments and marginal probabilities indexed by $I \subseteq D$ are identical. Higher order cross-moments coincide with first order cross-moments. The range of possible cross-moments is limited by the following constraints.

**Proposition 3.2.1.** *The cross-moments of binary data fulfill the sharp inequalities*

$$\max \left\{ \sum_{i \in I} m_i - |I| + 1, 0 \right\} \leq m_I \leq \min \{ m_K \colon K \subseteq I \}. \tag{3.1}$$

*Proof.* The lower bound follows from

$$|I| - 1 = \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} (|I| - 1) \pi(\boldsymbol{\gamma}) \geq \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} \left( \sum_{i \in I} \gamma_i - \prod_{i \in I} \gamma_i \right) \pi(\boldsymbol{\gamma}) = \sum_{i \in I} m_i - m_I,$$

the upper bound is the monotonicity of the measure. □

For the special case $|I| = 2$, Proposition 3.1 is a well-known result and has been invoked in several articles dealing with correlated binary data. For the general case, we remark that a mapping

$$f \colon [0,1]^{|I|} \to [0,1], \quad f_I(m_{i_1}, \ldots, m_{i_{|I|}}) = m_I,$$

which assigns a cross-moment $m_I$ for $I \subseteq D$ as function of the marginals $m_i$ for $i \in I$, is quite similar to a $|I|$-dimensional copula and the inequalities (3.1) are exactly the Fréchet-Hoeffding bounds (Nelsen, 2006, ch. 2).

**Definition 3.2.2.** We say a $d \times d$ symmetric matrix $\mathbf{M} := (m_{ij})$ with entries in $(0, 1)$ is a *cross-moment matrix of binary data* if $\mathbf{M} - \text{diag}(\mathbf{M})\text{diag}(\mathbf{M})^\intercal$ is positive definite and condition (3.1) holds for all $I \subseteq D$ with $|I| = 2$.

In the sequel we see how the cross-moment matrix relates to the notion of correlation.

**Definition 3.2.3.** For a set $I \subseteq D$, we define

$$u_I^\pi(\boldsymbol{\gamma}) := \prod_{i \in I} (\gamma_i - m_i^\pi)[m_i^\pi(1 - m_i^\pi)]^{-1/2},$$

and refer to $c_I^\pi := \mathbb{E}_\pi\left(u_I^\pi(\boldsymbol{X})\right)$ as the (generalized) correlation coefficient indexed by $I$.

A $d \times d$ positive definite matrix $\mathbf{C}$ with entries in $[-1, 1]$ and $\text{diag}(\mathbf{C}) = \mathbf{1}$ is not the correlation matrix of a binary distribution for every mean vector $\boldsymbol{m} \in (0, 1)^d$. In fact, $\mathbf{C}$ is a correlation matrix if and only if $\mathbf{M} = \mathbf{C} \cdot \boldsymbol{s}\boldsymbol{s}^\intercal + \boldsymbol{m}\boldsymbol{m}^\intercal$ is valid in the sense of Definition 3.2.2, where the dot means point-wise multiplication and $s_i^2 := m_i(1 - m_i)$. Chaganty and Joe (2006) elaborate alternative conditions for compatibility between correlations and means, but these do not seem easier to express or to check.

In the context of binary data, the notion of "strong correlations" refers to correlation coefficients which are at the boundary of the feasible range with respect to the mean vector. Note that the absolute value of the correlation coefficient does, in itself, not tell whether the correlation is easy or difficult to model. The following statement relates the notions of uncorrelated and independent variables.

**Proposition 3.2.2.** *Let $\boldsymbol{X}$ be a $d$-dimensional binary random vector. For $d = 2$, entries are uncorrelated if and only if they are independent. For $d \geq 3$, entries might be mutually uncorrelated but not independent.*

*Proof.* Let $p_{x_1 x_2} := \mathbb{P}(X_1 = x_1, X_2 = x_2)$. By definition $p_{11} = m_{12} = m_1 m_2$. Further, we obtain $p_{10} = m_1 - m_{12} = m_1(1 - m_2)$ and, analogously, $p_{01} = (1 - m_1)m_2$. Finally, we have $p_{00} = 1 + m_{12} - m_1 - m_2 = (1 - m_1)(1 - m_2)$. For $d \geq 3$, let for instance $p_{000} = p_{011} = p_{101} = p_{110} = 1/4$ and $p_{100} = p_{010} = p_{001} = p_{111} = 0$. The entries are mutually uncorrelated, but not independent since $p_{111} = 0 \neq 1/8 = m_1 m_2 m_3$. □

For some applications, it suffices to model structured dependencies, such as exchangeable ($c_{ij} = c$), moving average ($c_{ij} = c\mathbb{1}_{|i-j|=1}$) or autoregressive ($c_{ij} = c^{|i-j|}$) correlations for $i \neq j \in D$. There is a long series of articles concerned with efficient approaches to sampling binary vectors for structured correlations (Farrell and Sutradhar, 2006; Qaqish, 2003; Oman and Zucker, 2001; Lunn and Davies, 1998; Park et al., 1996). However, we focus on the problem of sampling binary data with arbitrary cross-moment matrix $\mathbf{M}$ which is a building block of general adaptive Monte Carlo algorithms.

### 3.2.2. Representations and bounds

**Proposition 3.2.3.** *Let $f\colon \mathbb{B}^d \to \mathbb{R}$ be some function and $\tau\colon \mathbb{R} \supseteq V \to \pi(\mathbb{B}^d)$ a bijective mapping. There are coefficients $a_I \in \mathbb{R}$ such that*

$$f(\boldsymbol{\gamma}) = \tau\left[\textstyle\sum_{I\subseteq D} a_I \prod_{i\in I} \gamma_i\right].$$

*Proof.* We denote by $\mathbb{1}(I) := (\mathbb{1}_I(1), \ldots, \mathbb{1}_I(d)) \in \mathbb{B}^d$ the indicator vector of the index set $I$. We thus have $f(\boldsymbol{\gamma}) = \tau[\sum_{I\subseteq D} \delta_{\mathbb{1}(I)}(\boldsymbol{\gamma})\tau^{-1}(f[\mathbb{1}(I)])]$ and writing the Dirac delta function as a product $\delta_{\mathbb{1}(I)}(\boldsymbol{\gamma}) = \prod_{i\in I} \gamma_i \prod_{i\in D\setminus I}(1-\gamma_i)$ we conclude the assertion. □

In particular, every binary distribution admits a multi-linear representation. The usefulness of this result is limited, however, since the coefficients of the expansion do not easily relate to the notion of cross-moments. However, the following representation by Bahadur (1961) allows to write a binary distribution in terms of its generalized correlation coefficients.

**Proposition 3.2.4.** *Let $\pi$ be a binary distribution with mean $\boldsymbol{m} \in (0,1)^d$. Then,*

$$\pi(\boldsymbol{\gamma}) = q^{\square}_{\boldsymbol{m}}(\boldsymbol{\gamma})\ \left[\textstyle\sum_{I\subseteq D} c^{\pi}_I u^{\pi}_I(\boldsymbol{\gamma})\right].$$

*Proof.* We give the proof by Bahadur (1961) using the notation introduced above. The set $\{u^{\pi}_I\colon I \subseteq D\}$ forms an orthonormal basis on $\mathcal{F} := \{f\colon \mathbb{B}^d \to \mathbb{R}\}$ with respect to the inner product

$$(f,g) = \mathbb{E}_{q^{\square}_{\boldsymbol{m}}}\left(f(\boldsymbol{X})g(\boldsymbol{X})\right) = \textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} f(\boldsymbol{\gamma})g(\boldsymbol{\gamma})q^{\square}_{\boldsymbol{m}}(\boldsymbol{\gamma}).$$

Therefore, every function $f \in \mathcal{F}$ has a unique representation $f(\boldsymbol{\gamma}) = \sum_{I\subseteq D}(f, u^{\pi}_I)u^{\pi}_I(\boldsymbol{\gamma})$. Compute the inner products

$$(\pi/q^{\square}_{\boldsymbol{m}}, u^{\pi}_I) = \textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d}[\pi(\boldsymbol{\gamma})/q^{\square}_{\boldsymbol{m}}(\boldsymbol{\gamma})]u^{\pi}_I(\boldsymbol{\gamma})q^{\square}_{\boldsymbol{m}}(\boldsymbol{\gamma}) = \mathbb{E}_{\pi}\left(u^{\pi}_I(\boldsymbol{X})\right) = c^{\pi}_I$$

to obtain the desired form $\pi(\boldsymbol{\gamma})/q^{\square}_{\boldsymbol{m}}(\boldsymbol{\gamma}) = \sum_{I\subseteq D} c^{\pi}_I u^{\pi}_I(\boldsymbol{\gamma})$. □

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 family times a correction term $1+\sum_{I\in\mathcal{I}_k} v_I(\boldsymbol{\gamma})\, c_I$ where the coefficients are higher order correlations.

Using Proposition 3.2.4, we may bound the $l^p$ distance between two binary distribution with the same mean in terms of nearness of their correlation coefficients.

**Proposition 3.2.5.** *Let $\pi$ and $\omega$ be binary distributions with mean $\boldsymbol{m} \in (0,1)^d$. For an exponent $p \geq 1$,*

$$\textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} |\pi(\boldsymbol{\gamma}) - \omega(\boldsymbol{\gamma})|^p \leq \sum_{I\subseteq D} 2^{(1-\min\{p,2\})|I|} |c_I^\pi - c_I^\omega|^p \leq (1+r)^d - dr - 1,$$

*where $r = 2^{1-\min\{p,2\}} \max_{I\subseteq D} |c_I^\pi - c_I^\omega|^{p/|I|}$.*

*Proof.* Since $u_I^\pi = u_I^\omega$ for all $I \subseteq D$, applying Proposition 3.2.4 yields

$$\begin{aligned}\textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^{bvsd}} |\pi(\boldsymbol{\gamma}) - \omega(\boldsymbol{\gamma})|^p &= \textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} \left| q_{\boldsymbol{m}}^{\square}(\boldsymbol{\gamma}) \sum_{I\subseteq D} u_I^\pi(\boldsymbol{\gamma})(c_I^\pi - c_I^\omega)\right|^p \\ &\leq \textstyle\sum_{I\subseteq D} |c_I^\pi - c_I^\omega|^p \, \mathbb{E}_{q_{\boldsymbol{m}}^{\square}} \left(|u_I^\pi(\boldsymbol{X})|^p\right).\end{aligned}$$

Using that $x^{p-1} + (1-x)^{p-1} \leq 2^{2-\min\{p,2\}}$ for all $x \in (0,1)$, we obtain the bound

$$\mathbb{E}_{q_{\boldsymbol{m}}^{\square}} \left(|u_I^\pi(\boldsymbol{X})|^p\right) \leq \textstyle\prod_{i\in I} [m_i(1-m_i)]^{1/2} [m_i^{p-1} + (1-m_i)^{p-1}] \leq 2^{(1-\min\{p,2\})|I|}.$$

Finally, we have $\sum_{I\subseteq D} 2^{(1-\min\{p,2\})|I|} |c_I^\pi - c_I^\omega|^p \leq \sum_{I\subseteq D, |I|\geq 2} r^{|I|} = (1+r)^d - dr - 1$, since by definition $c_I^\pi = c_I^\omega$ for all $I \subseteq D$ with $|I| \leq 2$. □

**Corollary 3.2.6.** *Let $\pi$ and $q$ be binary distributions with mean $\boldsymbol{m} \in (0,1)^d$. The total variation distance between $\pi$ and $q$ is bounded by $\frac{1}{2}\sum_{I\subseteq D} |c_I^\pi - c_I^q|$.*

**Proposition 3.2.7.** *Let $\pi$ and $q$ be binary distributions with cross-moment matrix $\mathbf{M}$. Then we have $\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} |\pi(\boldsymbol{\gamma}) - q(\boldsymbol{\gamma})|^p \leq (1+r)^d - \frac{1}{2}d(d-1)r^2 - dr - 1$.*

*Proof.* Analogously to Proposition 3.2.5. □

The last results merit a comment with regard to adaptive Monte Carlo algorithms. The summand $\frac{1}{2}d(d-1)r^2$ we have in Proposition 3.2.7 but not in Proposition 3.2.5 might be interpreted as the gain in "closeness" of the proposal to the target distribution when we compare a simple product model $q_{\boldsymbol{m}}^{\square}$ with $\boldsymbol{m} = \boldsymbol{m}^\pi = \boldsymbol{m}^q$ and a more sophisticated proposal distribution $q_{\mathbf{M}}$ with $\mathbf{M} = \mathbf{M}^\pi = \mathbf{M}^q$. In the following result, we formalize how the cross-moments of the proposal distribution affect the auto-covariance of the Metropolis-Hastings independence sampler. This underpins the practical observation that a proposal distribution which just matches the mean of the target distribution is often not flexible enough to yield an efficient Markov kernel.

**Proposition 3.2.8.** *Let $\pi$ and $q$ be binary distributions with mean $\boldsymbol{m} \in (0,1)^d$ and denote by $\kappa(\boldsymbol{\gamma} \mid \boldsymbol{x}) := q(\boldsymbol{\gamma})\lambda_q(\boldsymbol{\gamma},\boldsymbol{x}) + \delta_{\boldsymbol{x}}(\boldsymbol{\gamma})[1 - \sum_{\boldsymbol{y}\in\mathbb{B}^d} q(\boldsymbol{y})\lambda_q(\boldsymbol{y},\boldsymbol{x})]$ the Metropolis-Hastings kernel with invariant measure $\pi$ and proposal distribution $q$ where $\lambda_q(\cdot,\boldsymbol{x})$ is defined in (1.7). The auto-covariance between $(\boldsymbol{X}_1, \boldsymbol{X}_2) \sim \pi\kappa$ is*

$$\mathbb{E}_{\kappa\pi}\left(\boldsymbol{X}_2\boldsymbol{X}_1^\intercal\right) - \boldsymbol{m}\boldsymbol{m}^\intercal = \frac{1}{2}(\mathbf{M}^\pi - \mathbf{M}^q) + \mathbf{R}$$

*with $\mathbf{R} = (r_{ij})$ where $|r_{ij}| \leq \sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} |\pi(\boldsymbol{\gamma}) - q(\boldsymbol{\gamma})|$.*

*Proof.* We plug the definition of the kernel into the expected value and obtain

$$\begin{aligned}\mathbb{E}_{\pi\kappa}\left(\boldsymbol{X}_2\boldsymbol{X}_1^\intercal\right) &= \sum_{\boldsymbol{\gamma},\,\boldsymbol{x}\in\mathbb{B}^d} \gamma_i x_j \kappa(\boldsymbol{\gamma}\mid\boldsymbol{x})\pi(\boldsymbol{x}) \\ &= \sum_{\boldsymbol{\gamma},\,\boldsymbol{x}\in\mathbb{B}^d} \gamma_i x_j q(\boldsymbol{\gamma})\lambda_q(\boldsymbol{\gamma},\boldsymbol{x})\pi(\boldsymbol{x}) + \sum_{\boldsymbol{x}\in\mathbb{B}^d} x_i x_j [1-\textstyle\sum_{\boldsymbol{y}\in\mathbb{B}^d} q(\boldsymbol{y})\lambda_q(\boldsymbol{y},\boldsymbol{x})]\pi(\boldsymbol{x}) \\ &= m_{ij}^\pi + \sum_{\boldsymbol{\gamma},\,\boldsymbol{x}\in\mathbb{B}^d} (\gamma_i x_j - x_i x_j) q(\boldsymbol{\gamma})\pi(\boldsymbol{x})\lambda_q(\boldsymbol{\gamma},\boldsymbol{x}) \\ &= m_i m_j + \frac{1}{2}(m_{ij}^\pi - m_{ij}^q) + \frac{1}{2}\sum_{\boldsymbol{\gamma},\,\boldsymbol{x}\in\mathbb{B}^d} (\gamma_i x_j - x_i x_j)\left|q(\boldsymbol{\gamma})\pi(\boldsymbol{x}) - q(\boldsymbol{x})\pi(\boldsymbol{\gamma})\right|,\end{aligned}$$

where we used $2q(\boldsymbol{\gamma})\pi(\boldsymbol{x})\lambda_q(\boldsymbol{\gamma},\boldsymbol{x}) = q(\boldsymbol{\gamma})\pi(\boldsymbol{x}) + q(\boldsymbol{x})\pi(\boldsymbol{\gamma}) - |q(\boldsymbol{\gamma})\pi(\boldsymbol{x}) - q(\boldsymbol{x})\pi(\boldsymbol{\gamma})|$. The triangle inequality

$$\begin{aligned}&\sum_{\boldsymbol{\gamma},\,\boldsymbol{x}\in\mathbb{B}^d} |q(\boldsymbol{\gamma})\pi(\boldsymbol{x}) - q(\boldsymbol{x})\pi(\boldsymbol{\gamma})| = \sum_{\boldsymbol{\gamma},\,\boldsymbol{x}\in\mathbb{B}^d} |q(\boldsymbol{\gamma})\pi(\boldsymbol{x}) - \pi(\boldsymbol{\gamma})\pi(\boldsymbol{x}) + \pi(\boldsymbol{\gamma})\pi(\boldsymbol{x}) - q(\boldsymbol{x})\pi(\boldsymbol{\gamma})| \\ \leq &\sum_{\boldsymbol{\gamma},\,\boldsymbol{x}\in\mathbb{B}^d} \left[|q(\boldsymbol{\gamma}) - \pi(\boldsymbol{\gamma})|\,\pi(\boldsymbol{x}) + |\pi(\boldsymbol{x}) - q(\boldsymbol{x})|\,\pi(\boldsymbol{\gamma})\right] = 2\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} |\pi(\boldsymbol{\gamma}) - q(\boldsymbol{\gamma})|\,.\end{aligned}$$

yields the bound on $r_{ij} := \frac{1}{2}\sum_{\boldsymbol{\gamma},\,\boldsymbol{x}\in\mathbb{B}^d}(\gamma_i x_j - x_i x_j)\,|q(\boldsymbol{\gamma})\pi(\boldsymbol{x}) - q(\boldsymbol{x})\pi(\boldsymbol{\gamma})|$. □

For a proposal distribution $q_{\mathbf{M}}$ with $\mathbf{M} = \mathbf{M}^\pi = \mathbf{M}^q$, the auto-covariance first term vanishes and the remainders $|r_{ij}|$ are, on average, smaller as implied by Proposition 3.2.7.

## 3.3. Families based on generalized linear models

### 3.3.1. Definition

We want to construct a parametric family $q$ for sampling independent random vectors with specified dependencies. Sampling in high dimensions, however, requires the computation of conditional distributions $q(\gamma_i \mid \boldsymbol{\gamma}_{1:i-1})$, and it is therefore convenient to define the parametric family directly in terms of its conditionals.

**Definition 3.3.1.** Let $\mu\colon \overline{\mathbb{R}} \to [0,1]$ be a monotonic function and $\mathbf{A} := (a_{ij})$ a $d \times d$ real-valued lower triangular matrix. We refer to

$$q_{\mathbf{A}}^\mu(\boldsymbol{\gamma}) = \textstyle\prod_{i=1}^d \left[\mu(a_{ii} + \sum_{j=1}^{i-1} a_{ij}\gamma_j)\right]^{\gamma_i} \left[1 - \mu(a_{ii} + \sum_{j=1}^{i-1} a_{ij}\gamma_j)\right]^{1-\gamma_i},$$

as the $\mu$-conditionals family. By construction, it is easy to sample $\boldsymbol{x} \sim q_{\mathbf{A}}^\mu$ and evaluate $q_{\mathbf{A}}^\mu(\boldsymbol{x})$ point-wise, see Procedure 7.

**Algorithm 7:** Sampling via chain rule factorization

$\boldsymbol{x} = (0, \ldots, 0),\ p \leftarrow 1$
**for** $i = 1, \ldots, d$ **do**
  $c \leftarrow q^{\mu}_{\mathbf{A}}(x_i = 1 \mid \boldsymbol{x}_{1:i-1}) = \mu(a_{ii} + \sum_{j=1}^{i-1} a_{ij} x_j)$
  $u \leftarrow U \sim \mathcal{U}_{[0,1]}$
  **if** $u < c$ **then** $x_i \leftarrow 1$
  $p \leftarrow \begin{cases} p \cdot c & \textbf{if} \quad x_i = 1 \\ p \cdot (1 - c) & \textbf{if} \quad x_i = 0 \end{cases}$
**end**
**return** $\boldsymbol{x},\ p$

**Proposition 3.3.1.** *Let* $\mu \colon \overline{\mathbb{R}} \to [0, 1]$ *be a bijection and* $\boldsymbol{m} \in (0, 1)^d$ *a mean vector. For* $\mathbf{A} = \operatorname{diag}[\mu^{-1}(\boldsymbol{m})]$ *we have* $q^{\mu}_{\mathbf{A}} = q^{\sqcap}_{\boldsymbol{m}}$.

Qaqish (2003) discusses the $\mu$-conditionals family with a truncated linear link function $\mu(x) = \min\{\max\{x, 0\}, 1\}$. The linear structure allows to compute the parameters by simple matrix inversion; on the downside, the linear function is truncated and fails to accommodate complicated correlation structures; see Section 3.6 for a numerical comparison. Qaqish (2003) elaborates on conditions that guarantee the linear conditionals family to be valid for special correlation structures.

Farrell and Sutradhar (2006) propose a $\mu$-conditionals family with a logistic link function $\mu(x) = 1/[1 + \exp(-x)]$. However, they only analyze the special case of autoregressive correlation structure. The idea to model conditional probabilities by logistic regression terms has also been suggested by Arnold (1996). In Section 3.5.1, we further motivate the use of the logistic link function. In the following theorem, we formalize the fact that this approach indeed allows to model any feasible combination of mean and correlation structure.

**Theorem 3.3.2.** *Let* $\mu \colon \overline{\mathbb{R}} \to [0, 1]$ *be an increasing, differentiable bijection and* $\mathbf{M}$ *a* $d \times d$ *cross-moment matrix. There is a unique* $d \times d$ *real-valued lower triangular matrix* $\mathbf{A}$ *such that* $\sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} q^{\mu}_{\mathbf{A}}(\boldsymbol{\gamma}) \boldsymbol{\gamma} \boldsymbol{\gamma}^{\intercal} = \mathbf{M}$.

Popular link functions that verify the condition include the logistic function with $\mu(x) = 1/[1 + \exp(-x)]$, the probit function with $\mu(x) = (2\pi)^{-1/2} \int_{-\infty}^{x} \exp(-y^2/2) dy$, the arctan function $\mu(x) = 1/2 + \arctan(x)/\pi$ and the complementary log-log function $\mu(x) = 1 - \exp[-\exp(x)]$, see McCullagh and Nelder (1989, sec. 4.3). We derive two auxiliary results to structure the proof of Theorem 3.3.2.

**Lemma 3.3.3.** *For a cross-moment matrix* $\mathbf{M}$ *with mean vector* $\boldsymbol{m} = \operatorname{diag}(\mathbf{M})$*, we have*

$$\begin{pmatrix} \mathbf{M} & \boldsymbol{m} \\ \boldsymbol{m}^\intercal & 1 \end{pmatrix} > 0.$$

*Proof.* Note that $\boldsymbol{m}^\intercal \mathbf{M}^{-1}\boldsymbol{m} - (\boldsymbol{m}^\intercal \mathbf{M}^{-1}\boldsymbol{m})^2 = (\mathbf{M}^{-1}\boldsymbol{m})^\intercal(\mathbf{M} - \boldsymbol{m}\boldsymbol{m}^\intercal)\mathbf{M}^{-1}\boldsymbol{m} > 0$ because the covariance matrix $\mathbf{M} - \boldsymbol{m}\boldsymbol{m}^\intercal$ is positive definite. Dividing by $\boldsymbol{m}^\intercal \mathbf{M}^{-1}\boldsymbol{m} > 0$ we obtain $1 - \boldsymbol{m}^\intercal \mathbf{M}^{-1}\boldsymbol{m} > 0$ which yields

$$\begin{aligned} \det \begin{pmatrix} \mathbf{M} & \boldsymbol{m} \\ \boldsymbol{m}^\intercal & 1 \end{pmatrix} &= \det \left[ \begin{pmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0}^\intercal & 1 \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{M}^{-1}\boldsymbol{m} \\ \boldsymbol{m}^\intercal & 1 \end{pmatrix} \right] \\ &= \det(\mathbf{M}) \det \begin{pmatrix} \mathbf{I} & \mathbf{M}^{-1}\boldsymbol{m} \\ \mathbf{0}^\intercal & (1 - \boldsymbol{m}^\intercal \mathbf{M}^{-1}\boldsymbol{m}) \end{pmatrix} \\ &= \det(\mathbf{M})(1 - \boldsymbol{m}^\intercal \mathbf{M}^{-1}\boldsymbol{m}) > 0. \end{aligned}$$

Therefore, all principal minors are positive. □

**Lemma 3.3.4.** *Let* $\mu \colon \overline{\mathbb{R}} \to [0,1]$ *be a monotonic, differentiable bijection, and denote by* $B_r^n = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^\intercal \boldsymbol{x} < r^2\}$ *the open ball with radius* $r > 0$*. Let* $\pi$ *be a binary distribution with cross-moment matrix* $\mathbf{M}$*. We write* $\boldsymbol{m} = \operatorname{diag}(\mathbf{M})$ *and* $\boldsymbol{m}^* = (\boldsymbol{m}^\intercal, 1)^\intercal$ *for the mean vector. There is* $\varepsilon_r > 0$ *such that the function*

$$f \colon B_r^{d+1} \to \bigtimes_{i=1}^{d+1} (\varepsilon_r, m_i^* - \varepsilon_r), \quad f(\boldsymbol{a}) = \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} \pi(\boldsymbol{\gamma}) \mu(a_{d+1} + \textstyle\sum_{k=1}^d a_k \gamma_k) \begin{pmatrix} \boldsymbol{\gamma} \\ 1 \end{pmatrix}$$

*is a differentiable bijection.*

*Proof.* We set $\varepsilon_r := \max \bigcup_{i \in D \cup \{d+1\}} \left\{ \min_{\boldsymbol{a} \in B_r^{d+1}} f_i(\boldsymbol{a}),\ m_i^* - \max_{\boldsymbol{a} \in B_r^{d+1}} f_i(\boldsymbol{a}) \right\}$. For indices $i, j \in D \cup \{d+1\}$, the partial derivatives of $f$ are

$$\frac{\partial f_i}{\partial a_j} = \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} \pi(\boldsymbol{\gamma}) \mu'(a_{d+1} + \textstyle\sum_{k=1}^d a_k \gamma_k) \times \begin{cases} \gamma_i \gamma_j & (i, j \in \{1, \dots, d\}) \\ \gamma_i & (j = d+1) \\ \gamma_j & (i = d+1) \\ 1 & (i = j = d+1). \end{cases}$$

We have $\eta_r := \min_{\boldsymbol{a} \in B_r^{d+1}} \min_{\boldsymbol{\gamma} \in \mathbb{B}^d} \mu'(a_{d+1} + \sum_{i=1}^d a_i \gamma_i) > 0$ since $\mu$ is strictly increasing. Then the Jacobian is positive for all $\boldsymbol{a} \in B_r^d$,

$$\det f'(\boldsymbol{a}) = \det \left[ \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} \pi(\boldsymbol{\gamma}) \mu'(a_{d+1} + \textstyle\sum_{i=1}^d a_i \gamma_i) \begin{pmatrix} \boldsymbol{\gamma}\boldsymbol{\gamma}^\intercal & \boldsymbol{\gamma} \\ \boldsymbol{\gamma}^\intercal & 1 \end{pmatrix} \right] \geq \eta_r^{d+1} \det \begin{pmatrix} \mathbf{M} & \boldsymbol{m} \\ \boldsymbol{m}^\intercal & 1 \end{pmatrix} > 0,$$

where we applied Lemma 3.3.3 in the last inequality. □

*Proof of Theorem 3.3.2.* We proceed by induction over $d$. For $d = 1$, $\mathbf{A}(1)$ is a scalar and we define the $\mu$-conditionals family $q^{\mu}_{\mathbf{A}(1)}$ via Corollary 3.3.1. Suppose that we have already constructed a $\mu$-conditionals family $q^{\mu}_{\mathbf{A}(d)}$ with $d \times d$ lower triangular matrix $\mathbf{A}(d)$ and cross-moment matrix $\mathbf{M}(d)$. We can add a new dimension to the $\mu$-conditionals model $q^{\mu}_{\mathbf{A}(d)}$ without changing $\mathbf{M}(d)$, since

$$\begin{aligned}
\sum_{\boldsymbol{x}\in\mathbb{B}^{d+1}} q^{\mu}_{\mathbf{A}(d+1)}(\boldsymbol{x})\boldsymbol{x}\boldsymbol{x}^{\intercal} &= \sum_{\boldsymbol{x}\in\mathbb{B}^{d+1}} q^{\mu}_{\mathbf{A}(d)}(\boldsymbol{x}_{1:d})\boldsymbol{x}\boldsymbol{x}^{\intercal}\left[\mu(a_{d+1,d+1} + \textstyle\sum_{j=1}^{d} a_{d+1,j}x_j)\right]^{x_{d+1}} \times \\
&\qquad \left[1 - \mu(a_{d+1,d+1} + \textstyle\sum_{j=1}^{d} a_{d+1,j}x_j)\right]^{1-x_{d+1}} \\
&= \sum_{\boldsymbol{\gamma}\in\mathbb{B}^{d}} q^{\mu}_{\mathbf{A}(d)}(\boldsymbol{\gamma})\Bigg\{\mu(a_{d+1,d+1} + \textstyle\sum_{j=1}^{d} a_{d+1,j}\gamma_j)\begin{pmatrix}\boldsymbol{\gamma}\boldsymbol{\gamma}^{\intercal} & \boldsymbol{\gamma}\\ \boldsymbol{\gamma}^{\intercal} & 1\end{pmatrix} + \\
&\qquad \left[1 - \mu(a_{d+1,d+1} + \textstyle\sum_{j=1}^{d} a_{d+1,j}\gamma_j)\right]\begin{pmatrix}\boldsymbol{\gamma}\boldsymbol{\gamma}^{\intercal} & \mathbf{0}\\ \mathbf{0}^{\intercal} & 0\end{pmatrix}\Bigg\} \\
&= \sum_{\boldsymbol{\gamma}\in\mathbb{B}^{d}} q^{\mu}_{\mathbf{A}(d)}(\boldsymbol{\gamma})\mu(a_{d+1,d+1} + \textstyle\sum_{j=1}^{d} a_{d+1,j}\gamma_j)\begin{pmatrix}\mathbf{0} & \boldsymbol{\gamma}\\ \boldsymbol{\gamma}^{\intercal} & 1\end{pmatrix} + \\
&\qquad \begin{pmatrix}\mathbf{M}(d) & \mathbf{0}\\ \mathbf{0}^{\intercal} & 0\end{pmatrix}
\end{aligned}$$

For reasons of symmetry, it suffices to show that there is $\boldsymbol{a} \in \mathbb{R}^{d+1}$ such that

$$f(\boldsymbol{a}) = \sum_{\boldsymbol{\gamma}\in\mathbb{B}^{d}} q_{\mathbf{A}(d)}(\boldsymbol{\gamma})\mu(a_{d+1} + \textstyle\sum_{i=1}^{d} a_i\gamma_i)\begin{pmatrix}\boldsymbol{\gamma}\\ 1\end{pmatrix} = \mathbf{M}(d+1)_{\cdot,d+1},$$

where the r.h.s. denotes the $(d+1)$th column of the augmented cross-moment matrix. There is $\varepsilon > 0$ so that $\mathbf{M}(d+1)_{\cdot,d+1} \in \times_{i=1}^{d+1}(\varepsilon, m_i^* - \varepsilon)$ with $\boldsymbol{m}^* = (\mathrm{diag}[\mathbf{M}(d)]^{\intercal}, 1)$ which implies that a solution is contained in a sufficiently large open ball $B^{d+1}_{r_\varepsilon}$. We apply Lemma 3.3.4 to complete the inductive step and the proof. □

## 3.3.2. Maximum-likelihood

For a log-concave link function $\mu$, one can easily fit the $\mu$-conditionals family to weighted data $(\boldsymbol{w}, \mathbf{X})$ by component-wise likelihood maximization. We provide a review of likelihood maximization for generalized linear models with binary response in Section 5.2.1 in the context of Bayesian variable selection. Here, we only work out the explicit procedure for the special case of the logistic conditionals family since we advocate its use in the context of the SMC sampler developed in Chapter 2.

For an index $i \in D$, let $\boldsymbol{y}^{(i)} := \mathbf{X}_{\cdot,i}$ denote the vector of observations, $\mathbf{W} := \operatorname{diag}(\boldsymbol{w})$ a diagonal matrix with weights and $\mathbf{Z}^{(i)} := (\mathbf{X}_{\cdot,1:i-1}, \mathbf{1})$ the design matrix. The log-likelihood function for the weighted logistic regression is

$$\begin{aligned}\log l(\boldsymbol{a}) &= \sum_{k=1}^{n} w_k \left[ y_k^{(i)} \log[\ell(\boldsymbol{z}_{k,\cdot}^{(i)}\boldsymbol{a})] + (1 - y_k^{(i)}) \log[1 - \ell(\boldsymbol{z}_{k,\cdot}^{(i)}\boldsymbol{a})] \right] \\ &= \sum_{k=1}^{n} w_k \left[ y_k^{(i)} \boldsymbol{z}_{k,\cdot}^{(i)}\boldsymbol{a} - \log[1 + \exp(\boldsymbol{z}_{k,\cdot}^{(i)}\boldsymbol{a})] \right],\end{aligned}$$

where we used that $\log[1 - \ell(\boldsymbol{x}^\intercal\boldsymbol{a})] = -\log[1 + \exp(\boldsymbol{x}^\intercal\boldsymbol{a})] = -\boldsymbol{x}^\intercal\boldsymbol{a} + \log[\ell(\boldsymbol{x}^\intercal\boldsymbol{a})]$. Since $\partial \log[1 + \exp(\boldsymbol{x}^\intercal\boldsymbol{a})]/\partial\boldsymbol{a} = \ell(\boldsymbol{x}^\intercal\boldsymbol{a})\boldsymbol{x}$, the gradient of the log-likelihood is

$$s(\boldsymbol{a}) = \sum_{k=1}^{n} w_k \left[ y_k^{(i)} \boldsymbol{z}_{k,\cdot}^{(i)} - \ell(\boldsymbol{z}_{k,\cdot}^{(i)}\boldsymbol{a}) \boldsymbol{z}_{k,\cdot}^{(i)} \right] = (\mathbf{Z}^{(i)})^\intercal \mathbf{W} [\boldsymbol{y}^{(i)} - \boldsymbol{p}_{\boldsymbol{a}}^{(i)}],$$

where $(p_{\boldsymbol{a}}^{(i)})_k := \ell(\boldsymbol{z}_{k,\cdot}^{(i)}\boldsymbol{a})$. Since $\partial\ell(\boldsymbol{x}^\intercal\boldsymbol{a})/\partial\boldsymbol{a} = -\ell(\boldsymbol{x}^\intercal\boldsymbol{a})[1 - \ell(\boldsymbol{x}^\intercal\boldsymbol{a})]\boldsymbol{x}$, the observed Fisher information matrix is

$$F(\boldsymbol{a}) = \sum_{k=1}^{n} w_k \left[ \ell(\boldsymbol{z}_{k,\cdot}^{(i)}\boldsymbol{a})[1 - \ell(\boldsymbol{z}_{k,\cdot}^{(i)}\boldsymbol{a})] \right] \boldsymbol{z}_{k,\cdot}^{(i)} (\boldsymbol{z}_{k,\cdot}^{(i)})^\intercal = (\mathbf{Z}^{(i)})^\intercal \mathbf{W} \operatorname{diag}(\boldsymbol{q}_{\boldsymbol{a}}^{(i)}) \mathbf{Z}^{(i)},$$

where $q_{\boldsymbol{a},k}^{(i)} := \ell(\boldsymbol{z}_{k,\cdot}^{(i)}\boldsymbol{a})[1 - \ell(\boldsymbol{z}_{k,\cdot}^{(i)}\boldsymbol{a})]$. We put a normal prior $\mathcal{N}(\mathbf{0}, \varepsilon^{-1}\mathbf{I})$ on the regression parameters $\boldsymbol{a}$ to ensure that the likelihood function is convex, compare Section 5.2.1. The Newton Raphson iteration simplifies to $\boldsymbol{a}^{(t+1)} = \boldsymbol{a}^{(t)} + \boldsymbol{x}^{(t)}$ where $\boldsymbol{x}^{(t)}$ is the vector that solves $\left[ (\mathbf{Z}^{(i)})^\intercal \mathbf{W} \operatorname{diag}\left[\boldsymbol{q}_{\boldsymbol{a}^{(t)}}^{(i)}\right] \mathbf{Z}^{(i)} + \varepsilon\mathbf{I} \right] \boldsymbol{x}^{(t)} = (\mathbf{Z}^{(i)})^\intercal \mathbf{W} [\boldsymbol{y}^{(i)} - \boldsymbol{p}_{\boldsymbol{a}^{(t)}}^{(i)}] - \varepsilon\boldsymbol{a}^{(t)}$. We might choose $\boldsymbol{a}^{(0)} = (\mathbf{0}, \ell^{-1}(\overline{x}_i))$ as starting point, where $\overline{x}_i$ denotes the weighted sample mean. In the context of the SMC sampler discussed in Chapter 2, better initial values might be obtained from the parameter of the previous auxiliary distribution.

If the Newton iteration at the $i$th component fails to converge, we can either augment the penalty term $\varepsilon$ which leads to stronger shrinkage of the mean towards $1/2$ or we can drop some covariates $\gamma_j$ for $j \in [\![1, i-1]\!]$ from the iteration to improve the numerical condition of the procedure. In practice, we also drop the predictors from the regression model which are only weakly correlated with the explained variable, see Section 3.6.1. In particularly difficult cases, we might prefer to set $\boldsymbol{a} = (\mathbf{0}, \ell^{-1}(\overline{x}_i))$, where $\overline{x}_i$ denotes the weighted sample mean. This guarantees that at least the mean is correct which is important since misspecification of the mean of $\gamma_i$ obviously affects the distribution of the components $\gamma_j$ for $j \in [\![i+1, d]\!]$ which are sampled conditional on $\gamma_i$. Yet another way to tweak the numerical properties is re-parameterization through swapping the component $i$ and another component $j \in [\![i+1, d]\!]$. Later, we have to apply the inverse permutation in the sampling algorithm to deliver the binary vector in the original order.

**Algorithm 8:** ML fitting for a logistic conditionals family

**Input**: $\mathbf{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)^\intercal$, $\mathbf{W} = \operatorname{diag}(w_1, \dots, w_n)$, $\mathbf{A} \in \mathbb{R}^{d \times d}$

**for** $i \in D$ **do**

$\mathbf{Z} \leftarrow (\mathbf{X}_{\cdot,1:i-1}, \mathbf{1})$, $\boldsymbol{y} \leftarrow \mathbf{X}_{\cdot,i}$, $\boldsymbol{a}^{(0)} \leftarrow \mathbf{A}_{i,1:i}$

**repeat**

$p_k \leftarrow \ell(\mathbf{Z}_{k,\cdot}\boldsymbol{a}^{(t)})$ **for all** $k \in [\![1, n]\!]$

$q_k \leftarrow p_k(1 - p_k)$ **for all** $k \in [\![1, n]\!]$

$\boldsymbol{a}^{(t+1)} \leftarrow \boldsymbol{a}^{(t)} + \left[(\mathbf{Z}^{(i)})^\intercal \mathbf{W} \operatorname{diag}[\boldsymbol{q}] \mathbf{Z}^{(i)} + \varepsilon \mathbf{I}\right]^{-1} \left[(\mathbf{Z}^{(i)})^\intercal \mathbf{W} [\boldsymbol{y} - \boldsymbol{p}] - \varepsilon \boldsymbol{a}^{(t)}\right]$

**until** $\|\boldsymbol{a}^{(t+1)} - \boldsymbol{a}^{(t)}\|_\infty < \delta$

$\mathbf{A}_{i,1:i} \leftarrow \boldsymbol{a}$

**end**

**return A**

### 3.3.3. Method of moments

If we have data available instead of cross-moments, we rather fit a $\mu$-conditionals family via component-wise likelihood maximization than by method of moments since the former is faster and can even be parallelized, see Section 3.3.2. Still, in some applications we want to sample binary data with specified means and correlations, an example being the evaluation of statistical procedures for marginal regression models (Qaqish, 2003). Further, the practical range of cross-moments which can be sampled is a reasonable criterion to compare the flexibility of competing parametric families, and we use this for the numerical comparison in Section 3.6.

The proof of Theorem 3.3.2 suggests an iterative procedure to adjust the parameter $\mathbf{A}$ to a given cross-moment matrix $\mathbf{M}$. We add new cross-moments $\boldsymbol{m} \in (0, 1)^{d+1}$ to the $d \times d$ a lower triangular matrix $\mathbf{A}$ by solving the non-linear equation $f(\boldsymbol{a}) = \boldsymbol{m}$ via Newton-Raphson iterations $\boldsymbol{a}^{(k+1)} = \boldsymbol{a}^{(k)} - [f'(\boldsymbol{a}^{(k)})]^{-1}[f(\boldsymbol{a}^{(k)}) - \boldsymbol{m}]$ where

$$
\begin{aligned}
f(\boldsymbol{a}) &= \textstyle\sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} q^\mu_{\mathbf{A}}(\boldsymbol{\gamma}) \mu[(\boldsymbol{\gamma}^\intercal, 1)\boldsymbol{a}](\boldsymbol{\gamma}^\intercal, 1)^\intercal \\
f'(\boldsymbol{a}) &= \textstyle\sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} q^\mu_{\mathbf{A}}(\boldsymbol{\gamma}) \mu'[(\boldsymbol{\gamma}^\intercal, 1)\boldsymbol{a}](\boldsymbol{\gamma}^\intercal, 1)^\intercal(\boldsymbol{\gamma}^\intercal, 1)
\end{aligned}
$$

For dimensions $d > 10$, the exact computation of the expectations becomes expensive, and we replace $f$ and $f'$ by their Monte Carlo estimates

$$
\begin{aligned}
\hat{f}(\boldsymbol{a}) &= \textstyle\sum_{k=1}^n q^\mu_{\mathbf{A}}(\boldsymbol{\gamma}) \mu[(\boldsymbol{x}_k^\intercal, 1)\boldsymbol{a})](\boldsymbol{x}_k^\intercal, 1) \\
\hat{f}'(\boldsymbol{a}) &= \textstyle\sum_{k=1}^n q^\mu_{\mathbf{A}}(\boldsymbol{\gamma}) \mu'[(\boldsymbol{x}_k^\intercal, 1)\boldsymbol{a})](\boldsymbol{x}_k^\intercal, 1)^\intercal(\boldsymbol{x}_k^\intercal, 1)
\end{aligned}
\tag{3.2}
$$

where $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are drawn from $q^\mu_{\mathbf{A}}$. Some remarks are in order.

If the smallest eigenvalue of $\mathbf{M} - \operatorname{diag}(\mathbf{M})\operatorname{diag}(\mathbf{M})^\intercal$ approaches zero or a cross-moment $m_{ij}$ approaches the bounds (3.1), the parameter $a_{ij}$ may become very large

in absolute value. The limited numerical accuracy available on a computer inhibits sampling from such extreme cases. We might encounter non-convergence in the course of the fitting procedure. In order to handle these problems, we set

$$m_{ij}(\lambda_k) := \lambda_k m_{ij} + (1-\lambda_k) m_{ii} m_{jj}, \quad 0 = \lambda_1 < \cdots < \lambda_n = 1$$

for all $j = 1, \ldots, i-1$ and compute a sequence of solutions $\boldsymbol{a}(\lambda_k)$ to the sequence of cross-moments $\boldsymbol{m}(\lambda_k)$. We stop if the parameters fail to converge which ensures that the mean of the $\mu$-conditionals family is always $\operatorname{diag}(\mathbf{M})$.

For the special case of the linear link function $\mu(x) = x$, we obtain

$$f(\boldsymbol{a}) = \left[\sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} q^{\mu}_{\mathbf{A}}(\boldsymbol{\gamma}) (\boldsymbol{\gamma}^\intercal, 1)^\intercal (\boldsymbol{\gamma}^\intercal, 1)\right] \boldsymbol{a} = \begin{pmatrix} \mathbf{M} & \boldsymbol{m} \\ \boldsymbol{m}^\intercal & 1 \end{pmatrix} \boldsymbol{a}$$

which always has a solution by virtue of Lemma 3.3.3; to construct a mass function, however, we have to fall back to the truncated version $\mu(x) = \min\{\max\{x, 0\}, 1\}$, and the range of feasible cross-moments is hard to assess (Qaqish, 2003).

## 3.4. Families based on multivariate copulas

### 3.4.1. Definition

Instead of constructing a parametric family with explicit conditionals $q_\theta(\gamma_i \mid \boldsymbol{\gamma}_{1:i-1})$, we could sample from an auxiliary parametric family $\varphi_\theta$ on $\mathbb{R}^d$ which allows to compute the conditionals $\varphi_\theta(x_i \mid \boldsymbol{x}_{1:i-1})$.

**Definition 3.4.1.** For a vector $\boldsymbol{a} \in \mathbb{R}^d$ and a parametric family $\varphi_\theta$ on $\mathbb{R}^d$ we define the copula family

$$q^c_{\boldsymbol{a},\theta}(\boldsymbol{\gamma}) := \int_{\tau_{\boldsymbol{a}}^{-1}(\boldsymbol{\gamma})} \varphi_\theta(\boldsymbol{x}) d\boldsymbol{x}, \quad \tau_{\boldsymbol{a}}(\boldsymbol{x}) := \left(\mathbb{1}_{(-\infty, a_1]}(x_1), \ldots, \mathbb{1}_{(-\infty, a_d]}(x_d)\right).$$

We do not need to explicitly compute the copula, but, obviously, the range of dependencies achievable with $q^c_{\boldsymbol{a},\theta}$ depends on the flexibility of the family of copulas given through the underlying auxiliary parametric family. For all $I \subseteq D$, the marginals are

$$\begin{aligned} m^c_I &= \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} q^c_{\boldsymbol{a},\theta}(\boldsymbol{\gamma}) \prod_{i \in I} \gamma_i = \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d, \boldsymbol{\gamma}_I = \mathbf{1}} \int_{\tau_{\boldsymbol{a}}^{-1}(\boldsymbol{\gamma})} \varphi_\theta(\boldsymbol{v})\, d\boldsymbol{v} \\ &= \int_{\bigcup_{\boldsymbol{\gamma} \in \mathbb{B}^d, \boldsymbol{\gamma}_I = \mathbf{1}} \{\tau_{\boldsymbol{a}}^{-1}(\boldsymbol{\gamma})\}} \varphi_\theta(\boldsymbol{v})\, d\boldsymbol{v} = \int_{\times_{i=1}^d \begin{cases} (-\infty, a_i] & i \in I \\ (-\infty, \infty) & i \notin I \end{cases}} \varphi_\theta(\boldsymbol{v})\, d\boldsymbol{v}, \end{aligned}$$

which is the marginal cumulative distribution function of the auxiliary distribution.

For a $d \times d$ correlation matrix $\boldsymbol{\Sigma}$, Emrich and Piedmonte (1991) propose the multivariate normal distribution

$$\varphi^n_{\boldsymbol{\Sigma}}(\boldsymbol{x}) = (2\pi)^{-d/2} \, |\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\tfrac{1}{2} \, \boldsymbol{x}^\intercal \boldsymbol{\Sigma}^{-1} \boldsymbol{x}\right)$$

as auxiliary parametric family. Alternatively, we could use a multivariate student's t distribution

$$\varphi^t_{\boldsymbol{\Sigma}}(\boldsymbol{x}) = \Gamma([\nu + d]/2)[\Gamma(\nu/2)(\nu\pi)^{d/2} \, |\boldsymbol{\Sigma}|^{1/2} \, (1 + \tfrac{1}{\nu} \boldsymbol{x}^\intercal \boldsymbol{\Sigma}^{-1} \boldsymbol{x})]^{-(\nu+d)/2}.$$

The point-wise evaluation of $q^c_{\boldsymbol{a},\boldsymbol{\Sigma}}(\boldsymbol{\gamma})$ requires the computation of multivariate probabilities, that is high-dimensional integrals with the respect to the density of the multivariate normal or student's t distribution. This is a computationally challenging task in itself, for details see Genz and Bretz (2009), and the copula families are therefore not easily incorporated into the adaptive Monte Carlo algorithms which rely on Markov transitions since these require computation of the mass function up to a constant.

### 3.4.2. Further copula approaches

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 copulas: Lee (1993) describes how to construct an Archimedean copula, more precisely the Frank family (Nelsen, 2006, p.119), for sampling multivariate binary data.

We need to solve a non-linear equation for each component when sampling 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, there are faster methods which enumerate the solution space $\mathbb{B}^d$ and construct explicit probabilities (Gange, 1995) which allows to draw from an alias table (Walker, 1977).

### 3.4.3. Method of moments

Let $\Phi(x)$ denote the univariate and $\Phi(x_1, x_2, \sigma)$ the bivariate cumulative distribution functions of the underlying auxiliary distribution where $\sigma \in [-1, 1]$ is the correlation coefficient. We may evaluate the bivariate cumulative distribution functions using fast

series approximations; see Drezner and Wesolowsky (1990) for bivariate normal and Genz and Bretz (2002) for bivariate student's t distributions.

Given the cross-moments $\mathbf{M}$ with $\boldsymbol{m} = \text{diag}(\mathbf{M})$, we set $a_i = \Phi^{-1}(m_i)$ for $i \in D$ to adjust the mean. In order to compute the parameter $\boldsymbol{\Sigma}$ which yields the desired cross-moments, we solve

$$m_{ij} = \Phi(a_i, a_j, \sigma_{ij})$$

for $\sigma_{ij}$ via bisectional search for all $i, j \in D$ with $i < j$. The function $\Phi(a_i, a_j, \sigma)$ is strictly monotonic in $\sigma$ since for both the normal and the Student's t bivariate cumulutive distribution function, we easily verify $\partial\Phi(a_i, a_j, \sigma)/\partial\sigma > 0$. Modarres (2011) suggests the bivariate Plackett (1965) distribution as a proxy which might provide a good starting value $\sigma_{ij}^0 \in (-1, 1)$. In the sequential Monte Carlo context, better initial values might be provided by the parameter of the previous auxiliary distributions.

In the case of the normal copula family we might use the standard result on the derivative $\partial\Phi^n(a_1, a_2, \sigma)/\partial\sigma = \varphi^n(a_i, a_j, \sigma)$ (Johnson et al., 2002, p.255) and solve $m_{ij} = \Phi^n(a_i, a_j, \sigma_{ij})$ for $\sigma_{ij}$ via Newton-Raphson iterations; see Procedure 9. However, the bivariate integral approximations are critical when $\sigma$ comes very close to either boundary of $[-1, 1]$. The Newton iteration might repeatedly fail when restarted at the corresponding boundary $\sigma_{ij}^{(0)} \in \{-1, 1\}$, and we might need to fall back to bisectional search which is always feasible.

While we always obtain a solution in the bivariate case, it is well-known that the resulting matrix $\boldsymbol{\Sigma}$ is not necessarily positive definite due to the range of the elliptical copulas which allow to attain the bounds (3.1) for $d \leq 2$, but not for higher dimensions. In that case, we can replace $\boldsymbol{\Sigma}$ by

$$\boldsymbol{\Sigma}^* = (\boldsymbol{\Sigma} + |\lambda|\,\mathbf{I})/(1 + |\lambda|) > 0 \tag{3.3}$$

where $\lambda$ is smaller than any eigenvalue of $\boldsymbol{\Sigma}$. Alternatively, we can project $\boldsymbol{\Sigma}$ into the set of correlation matrices; see Higham (2002) and follow-up papers for algorithms that compute the nearest correlation matrix in Frobenius norm.

## 3.5. Families based on other techniques

### 3.5.1. Multiplicative interactions

Consider the family of distributions which, under the constraints that $\pi$ has given cross-moments, maximizes the entropy

$$H(\pi) := -\sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} \pi(\boldsymbol{\gamma}) \log[\pi(\boldsymbol{\gamma})].$$

**Algorithm 9:** Fitting the normal copula family

**Input**: $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\intercal$, $\boldsymbol{w} = (w_1, \ldots, w_n)$, $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$
$\overline{\mathbf{X}} \leftarrow \sum_{k=1}^{n} w_k \boldsymbol{x}_k \boldsymbol{x}_k^\intercal$
**for** $i \in D$ **do** $a_i \leftarrow \Phi^{-1}(\overline{x}_{ii})$
**for** $i, j \in D,\ i < j$ **do**
  **repeat**
    $\sigma_{ij}^{(t+1)} \leftarrow \sigma_{ij}^{(t)} - \dfrac{\Phi(a_i, a_j, \sigma_{ij}^{(t)}) - \overline{x}_{ij}}{\varphi(a_i, a_j, \sigma_{ij}^{(t)})}$
  **until** $|\sigma_{ij}^{(t+1)} - \sigma_{ij}^{(t)}| < \delta$
  $\sigma_{ji} \leftarrow \sigma_{ij}^{(t+1)}$
**end**
**if not** $\boldsymbol{\Sigma} > 0$ **then** $\boldsymbol{\Sigma} \leftarrow (\boldsymbol{\Sigma} + |\lambda|\,\mathbf{I})/(1 + |\lambda|)$
**return** $\boldsymbol{a}$, $\boldsymbol{\Sigma}$

The following proposition is just a special case of a more general concept (Soofi, 1994).

**Proposition 3.5.1.** *Let $\mathcal{I} \subseteq 2^D$ be a family of index sets such that $\{m_I : I \in \mathcal{I}\}$ is a valid set of cross-moments. The maximum entropy distribution having the specified cross-moments $m_I$ for $I \in \mathcal{I}$ has the form*

$$q(\boldsymbol{z}) = \exp(\nu + \textstyle\sum_{I \in \mathcal{I}} a_I \prod_{i \in I} \gamma_i).$$

*with normalizing constant* $\nu := -\log[\sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} \exp(\sum_{I \in \mathcal{I}} a_I \prod_{i \in I} \gamma_i)]$.

*Proof.* Define the Lagrange multipliers $L(\pi, \boldsymbol{a}) = \sum_{I \in \mathcal{I}} a_I [\sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} \pi(\boldsymbol{\gamma}) \prod_{i \in I} \gamma_i - m_I]$ and differentiate $\partial[H(\pi) + L(\pi, \boldsymbol{a})]/\partial\pi(\boldsymbol{\gamma}) = -\log[\pi(\boldsymbol{\gamma})] - 1 + \sum_{I \in \mathcal{I}} a_I \prod_{i \in I} \gamma_i$. Solving the first order condition and normalizing completes the proof. □

Maximum entropy solutions are a natural way to design parametric families. The binary versions link to information theory (Soofi, 1994), log-linear theory for contingency tables (Bishop et al., 1975, ch. 5) and graphical models (Cox and Wermuth, 1996, ch. 2). They also play a central role in physics and life science being the well-studied Ising model on a weighted complete graph.

**Definition 3.5.1.** Let $\mathbf{A}$ be a $d \times d$ real-valued lower triangular matrix. We refer to

$$q_{\mathbf{A}}^{e}(\boldsymbol{\gamma}) = \exp(\nu + \boldsymbol{\gamma}^\intercal \mathbf{A} \boldsymbol{\gamma}),$$

as the exponential quadratic family with $\nu := -\log[\sum_{\boldsymbol{x} \in \mathbb{B}^d} \exp(\boldsymbol{x}^\intercal \mathbf{A} \boldsymbol{x})]$.

**Proposition 3.5.2.** *If $\mathbf{A} = \operatorname{diag}(\boldsymbol{a})$, then $a_{ii} = \ell^{-1}(m_{ii})$ and $q_{\mathbf{A}}^{e} = q_{\mathbf{A}}^{\ell} = q_{\boldsymbol{m}}^{\square}$.*

The exponential quadratic family appears to be the binary analogue of the multivariate normal distribution which is the maximum entropy distribution on $\mathbb{R}^d$ having a specified covariance matrix (Kapur, 1989, sec. 5.1.1). Finding its mode is an NP-hard optimization problem and intensively studied in the field of operation research (Boros et al., 2007, for a recent review).

**Proposition 3.5.3.** *The marginal distribution of the exponential quadratic family is*

$$q^e_{\mathbf{A}}(\boldsymbol{\gamma}_{-i}) = \exp\left(\nu + \boldsymbol{\gamma}_{-i}^\intercal \mathbf{A}_{-i}\boldsymbol{\gamma}_{-i} + \log\left[1 + \exp(a_{ii} + \textstyle\sum_{j=1}^{i-1} a_{ij}\gamma_j + \sum_{j=i+1}^{d} a_{ji}\gamma_j)\right]\right).$$

*Proof.* Straightforward, since

$$\begin{aligned} q^e_{\mathbf{A}}(\boldsymbol{\gamma}_{-i}) &= q^e_{\mathbf{A}}(\gamma_i = 0, \boldsymbol{\gamma}_{-i}) + q^e_{\mathbf{A}}(\gamma_i = 1, \boldsymbol{\gamma}_{-i}) \\ &= \exp(\nu + \boldsymbol{\gamma}_{-i}^\intercal \mathbf{A}_{-i}\boldsymbol{\gamma}_{-i})\left[1 + \exp(a_{ii} + \textstyle\sum_{j=1}^{i-1} a_{ij}\gamma_j + \sum_{j=i+1}^{d} a_{ji}\gamma_j)\right]. \end{aligned}$$

□

**Proposition 3.5.4.** *The conditional distribution of the exponential quadratic family is*

$$q^e_{\mathbf{A}}(\gamma_i = 1 \mid \boldsymbol{\gamma}_{-i}) = \ell(a_{ii} + \textstyle\sum_{j=1}^{i-1} a_{ij}\gamma_j + \sum_{j=i+1}^{d} a_{ji}\gamma_j).$$

*where* $\ell(x) := 1/[1 + \exp(-x)]$ *is the logistic link function.*

*Proof.* Straightforward, since

$$q^e_{\mathbf{A}}(\gamma_i = 1 \mid \boldsymbol{\gamma}_{-i}) = \frac{\exp(\nu + \boldsymbol{\gamma}_{-i}^\intercal \mathbf{A}_{-i}\boldsymbol{\gamma}_{-i} + a_{ii} + \sum_{j=1}^{i-1} a_{ij}\gamma_j + \sum_{j=i+1}^{d} a_{ji}\gamma_j)}{\exp(\nu + \boldsymbol{\gamma}_{-i}^\intercal \mathbf{A}_{-i}\boldsymbol{\gamma}_{-i})\left[1 + \exp(a_{ii} + \sum_{j=1}^{i-1} a_{ij}\gamma_j + \sum_{j=i+1}^{d} a_{ji}\gamma_j)\right]}$$

□

**Definition 3.5.2.** Let $\boldsymbol{X} \sim q$ be a binary random vector. Define the conditional log odd ratios

$$\omega^q_{ij} := \log\left[\frac{\mathbb{P}(X_i = 1 \mid X_j = 1, \boldsymbol{X}_{-i,j})\,\mathbb{P}(X_i = 0 \mid X_j = 0, \boldsymbol{X}_{-i,j})}{\mathbb{P}(X_i = 0 \mid X_j = 1, \boldsymbol{X}_{-i,j})\,\mathbb{P}(X_i = 1 \mid X_j = 0, \boldsymbol{X}_{-i,j})}\right]$$

**Proposition 3.5.5.** *The exponential quadratic family has constant conditional log odd ratios* $\omega^{q^e_{\mathbf{A}}}_{ij} = a_{ij}$.

*Proof.* The log odd ratios can be written as

$$\omega^q_{ij} = \ell^{-1}[\mathbb{P}(X_i = 1 \mid X_j = 1, \boldsymbol{X}_{-i,j})] - \ell^{-1}[\mathbb{P}(X_i = 1 \mid X_j = 0, \boldsymbol{X}_{-i,j})],$$

and the result follows immediately from Proposition 3.5.4. □

We can therefore read the parameters $a_{ij}$ as Lagrange multipliers or, if $i \neq j$, as conditional log odd-ratios. The constant conditional log odd ratios are the binary analogue of the constant conditional correlations of the multivariate normal distribution (Wermuth, 1976).

### Logistic conditionals approximation

Despite the numerous similarities to the multivariate normal distribution, we cannot easily sample from the exponential quadratic family nor explicitly relate the parameter $\mathbf{A}$ to the cross-moment matrix $\mathbf{M}$. The reason is that the lower dimensional marginal distributions are difficult to compute (Cox, 1972, (iii)) since the multi-linear structure is lost. We denote by $q^{\ell}_{\mathbf{A}}$ the logistic conditionals family, that is the $\mu$-conditionals family with logistic link function $\ell(x) := 1/[1+\exp(-x)]$. The following result shows that the logistic conditionals family is precisely constructed such that the non-linear term in the marginals vanishes.

**Proposition 3.5.6.** *Let $\mathbf{A}$ be a $d \times d$ lower triangular matrix. The logistic conditionals family can be written as*

$$q^{\ell}_{\mathbf{A}}(\boldsymbol{\gamma}) = \exp\left(\boldsymbol{\gamma}^{\intercal}\mathbf{A}\boldsymbol{\gamma} - \textstyle\sum_{i=1}^{d} \log\left[1+\exp(a_{ii}+\sum_{j=1}^{i-1} a_{ij}\gamma_j)\right]\right).$$

*Proof.* Straightforward calculations yield

$$\begin{aligned}
\log q^{\ell}_{\mathbf{A}}(\boldsymbol{\gamma}) &= \textstyle\sum_{i=1}^{d} \log\left([\ell(a_{ii}+\sum_{j=1}^{i-1} a_{ij}\gamma_j)]^{\gamma_i}[1-\ell(a_{ii}+\sum_{j=1}^{i-1} a_{ij}\gamma_j)]^{1-\gamma_i}\right)\\
&= \textstyle\sum_{i=1}^{d} \left(\gamma_i \log[\ell(a_{ii}+\sum_{j=1}^{i-1} a_{ij}\gamma_j)] + (1-\gamma_i)\log[1-\ell(a_{ii}+\sum_{j=1}^{i-1} a_{ij}\gamma_j)]\right)\\
&= \textstyle\sum_{i=1}^{d} \left(\gamma_i\, \ell^{-1}[\ell(a_{ii}+\sum_{j=1}^{i-1} a_{ij}\gamma_j)] + \log[1-\ell(a_{ii}+\sum_{j=1}^{i-1} a_{ij}\gamma_j)]\right)\\
&= \textstyle\sum_{i=1}^{d} \left(\gamma_i(a_{ii}+\sum_{j=1}^{i-1} a_{ij}\gamma_j) - \log[1+\exp(a_{ii}+\sum_{j=1}^{i-1} a_{ij}\gamma_j)]\right)\\
&= \textstyle\sum_{i=1}^{d}\sum_{j=1}^{i} a_{ij}\gamma_i\gamma_j - \sum_{i=1}^{d}\log[1+\exp(a_{ii}+\sum_{j=1}^{i-1} a_{ij}\gamma_j)]\\
&= \boldsymbol{\gamma}^{\intercal}\mathbf{A}\boldsymbol{\gamma} - \textstyle\sum_{i=1}^{d}\log[1+\exp(a_{ii}+\sum_{j=1}^{i-1} a_{ij}\gamma_j)],
\end{aligned}$$

where we used $\log[1-\ell(x)] = -\log[1+\exp(x)]$ in the third line. □

Since we cannot repeat the marginalization for lower dimensions, we cannot assess the lower dimensional conditional probabilities which are necessary for sampling. We can, however, derive a series of approximate marginal probabilities that produce a logistic conditionals family which is, for low correlations, close to the original exponential quadratic family. This idea goes back to Cox and Wermuth (1994).

**Proposition 3.5.7.** *Let $c_1 + c_2 x + c_3 x^2 \approx \log[\cosh(x)]$ be a second order approximation. We may approximate the marginal distribution $q^{e}_{\mathbf{A}}(\boldsymbol{\gamma}_{-d})$ by an exponential quadratic family $\exp(\nu_* + \boldsymbol{\gamma}_{-d}^{\intercal}\mathbf{A}_*\boldsymbol{\gamma}_{-d})$ with parameters*

$$\nu_* := \nu + \log(2) + c_1 + \tfrac{1}{2}a_{dd}, \quad \mathbf{A}_* := \mathbf{A}_{-d} + (c_2+\tfrac{1}{2})\mathrm{diag}(\boldsymbol{a}_*) + c_3\, \boldsymbol{a}_*\boldsymbol{a}_*^{\intercal},$$

*where $\boldsymbol{a}_* := (a_{d1}, \ldots, a_{d\,d-1})^{\intercal}$ denotes the $d$th column of $\mathbf{A}$ without $a_{dd}$.*

*Proof.* We write the marginal distribution of the exponential quadratic family as

$$q^e_{\mathbf{A}}(\boldsymbol{\gamma}_{-d}) = \exp\Big[\nu + \boldsymbol{\gamma}_{-d}^\intercal \mathbf{A}_{-d}\boldsymbol{\gamma}_{-d} + \tfrac{1}{2}(a_{dd} + \boldsymbol{a}_*^\intercal\gamma_{-d}) + \log\big(2\cosh\big[\tfrac{1}{2}(a_{dd} + \boldsymbol{a}_*^\intercal\gamma_{-d})\big]\big)\Big].$$

using the identity

$$\log[1+\exp(x)] = \log\big(\exp(\tfrac{1}{2}x)\big[\exp(-\tfrac{1}{2}x) + \exp(\tfrac{1}{2}x)\big]\big) = \tfrac{1}{2}x + \log\big[2\cosh(\tfrac{1}{2}x)\big]$$

and approximate the non-quadratic term by the second order polynomial

$$\log[\cosh(\tfrac{1}{2}a_{dd} + \tfrac{1}{2}\boldsymbol{a}_*^\intercal\boldsymbol{\gamma}_{-d})] \approx c_1 + c_2\boldsymbol{a}_*^\intercal\boldsymbol{\gamma}_{-d} + c_3(\boldsymbol{a}_*^\intercal\boldsymbol{\gamma}_{-d})^2.$$

We rewrite the inner products $\boldsymbol{a}_*^\intercal\boldsymbol{\gamma}_{-d} + (\boldsymbol{a}_*\boldsymbol{\gamma}_{-d})^2 = \boldsymbol{\gamma}_{-d}^\intercal\left[\operatorname{diag}(\boldsymbol{a}_*) + \boldsymbol{a}_*\boldsymbol{a}_*^\intercal\right]\boldsymbol{\gamma}_{-d}$ and rearrange the quadratic terms. □

We can iterate the procedure to construct a logistic conditionals family which is close to the original exponential quadratic family. However, the function $\log[\cosh(x)]$ behaves like a quadratic function around zero and like the absolute value function for large $|x|$. Thus, a quadratic polynomial can only approximate $\log[\cosh(x)]$ well for small values of $x$ which means that exponential quadratic families with strong dependencies are hard to approximate.

Cox and Wermuth (1994) propose a Taylor approximation which fits well around $\frac{1}{2}a_{dd}$ and works for weak correlations. The parameters are

$$\boldsymbol{c} = \big(\log[\cosh(\tfrac{1}{2}a_{dd})]), \tfrac{1}{2}\tanh(\tfrac{1}{2}a_{dd}), \tfrac{1}{8}\operatorname{sech}^2(\tfrac{1}{2}a_{dd})\big).$$

Alternatively, we define sampling points $x_1, \ldots, x_n$, compute $y_k = \log\cosh(\frac{1}{2}a_{dd} + x_k)$ and use the least squares estimate

$$\boldsymbol{c} = [(\mathbf{1}, \boldsymbol{x}, \boldsymbol{x}^2)^\intercal(\mathbf{1}, \boldsymbol{x}, \boldsymbol{x}^2)]^{-1}(\mathbf{1}, \boldsymbol{x}, \boldsymbol{x}^2)\boldsymbol{y}.$$

This provides a better overall approximation, but the fit might be poor around $\frac{1}{2}a_{dd}$.

### 3.5.2. Additive interactions

Taking $\tau$ the identity mapping in Proposition 3.2.3, we obtain a multi-linear representation

$$\pi(\boldsymbol{\gamma}) = \textstyle\sum_{I\subseteq D} a_I \prod_{i\in I}\gamma_i,$$

but it seems hard to give a useful interpretation of the coefficients $a_I$. We can construct a more parsimonious family by removing higher order interaction terms. For additive interactions, however, we face the problem that truncated representations do not necessarily define probability distributions since they might be negative.

**Definition 3.5.3.** For a symmetric matrix $\mathbf{A}$ we define the additive linear family

$$q^a_{\mathbf{A},a_0}(\boldsymbol{\gamma}) = \nu(a_0 + \boldsymbol{\gamma}^\intercal \mathbf{A}\boldsymbol{\gamma}), \tag{3.4}$$

where $\nu := [2^d a_0 + \sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} \boldsymbol{\gamma}^\intercal \mathbf{A}\boldsymbol{\gamma}]^{-1}$ and $a_0 = (-\min_{\boldsymbol{\gamma}\in\mathbb{B}^d} \boldsymbol{\gamma}^\intercal \mathbf{A}\boldsymbol{\gamma}) \vee 0$.

This definition is of little practical value, however, since $a_0$ is the solution of NP-hard optimization problem, see Section 6.3.1. In virtue of the linear structure, we can derive polynomial expressions for the cross-moments and marginal distributions.

**Proposition 3.5.8.** *For a set of indices $I \subseteq D$, we can write the corresponding cross-moment as*

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

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

**Lemma 3.5.9.** *For a set $I \subseteq D$ of indices it holds that*

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

*Proof.* For an index set $M \subseteq D$, we have the sum formula $\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} \prod_{k\in M} \gamma_k = 2^{d-|M|}$. 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 3.5.9. □

**Lemma 3.5.10.**

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

*Proof.* Straightforward calculations yield

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

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

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

The last line is the assertion of Lemma 3.5.9. □

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

$$\begin{aligned}
m_I &= \textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d}\left(\prod_{k\in I}\gamma_k\right)\ \nu(a_0+\boldsymbol{\gamma}^\intercal\mathbf{A}\boldsymbol{\gamma}) \\
&= \nu\left[\textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d}a_0+\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d}\left(\prod_{k\in I}\gamma_k\right)\sum_{i\in D}\sum_{j\in D}\gamma_i\gamma_j\ a_{i,j}\right] \\
&= \nu\left[2^{d-|I|}a_0+\textstyle\sum_{i\in D}\sum_{j\in D}a_{i,j}\ \ \sum_{\boldsymbol{\gamma}\in\mathbb{B}^d}\left(\prod_{k\in I\cup\{i,j\}}\gamma_k\right)\right] \text{ (Lemma 3.5.9)} \\
&= \nu\left[2^{d-|I|}a_0+\ \textstyle\sum_{i\in D}\sum_{j\in D}2^{d-|I\cup\{i,j\}|}\ a_{i,j}\right] \\
&= \nu 2^{d-|I|-2}\left[4a_0+\textstyle\sum_{i\in D}\sum_{j\in D}2^{\mathbb{1}_I(i)+\mathbb{1}_{I\cup\{i\}}(j)}\ a_{i,j}\right] \text{ (Lemma 3.5.10)} \\
&= \nu 2^{d-|I|-2}\left[4a_0+\mathbf{1}^\intercal\mathbf{A}\mathbf{1}+\operatorname{tr}(\mathbf{A})+\textstyle\sum_{i\in I}\left[2\sum_{j\in D}a_{i,j}+\sum_{j\in I\setminus\{i\}}a_{i,j}\right]\right]
\end{aligned}$$

Since $m_\emptyset = 1$ by definition, we the normalizing constant is

$$\nu = 2^{-d+2}\left(4a_0+\mathbf{1}^\intercal\mathbf{A}\mathbf{1}+\operatorname{tr}(\mathbf{A})\right)^{-1},$$

which allows us to write down the normalized cross-moments

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

The proof is complete. □

**Corollary 3.5.11.** *The normalizing constant is*

$$\nu = 2^{-d+2}\left(4a_0+\mathbf{1}^\intercal\mathbf{A}\mathbf{1}+\operatorname{tr}(\mathbf{A})\right)^{-1},$$

*and the expected value is*

$$\mathbb{E}_{q^a_{\mathbf{A},a_0}}(\gamma_i) = \frac{1}{2}+\frac{\sum_{k=1}^d a_{i,k}}{4a_0+\mathbf{1}^\intercal\mathbf{A}\mathbf{1}+\operatorname{tr}(\mathbf{A})}.$$

The mean $m_i$ is close to $1/2$ unless the row $\boldsymbol{a}_i$ dominates the matrix. Therefore, if the parameter matrix $\mathbf{A}$ is non-negative definite, the marginal probabilities $m_i$ can hardly take values at the extremes of the unit interval. While being somewhat limited in the range of feasible parameters, the advantage of the additive linear family is that the marginal distributions are available in analytical form.

**Proposition 3.5.12.** *For the marginal distribution, it holds that*

$$q^{(1:k)}_{\mathbf{A},a_0}(\boldsymbol{\gamma}_{1:k}) = \textstyle\sum_{\boldsymbol{x}\in\mathbb{B}^{d-(k+1)}} q_{\mathbf{A},a_0}(\boldsymbol{\gamma}_{1:k},\boldsymbol{x}) = \nu 2^{d-k-2} s_k(\boldsymbol{\gamma}_{1:k}),$$

*where*

$$\begin{aligned} s_k(\boldsymbol{\gamma}_{1:k}) = 4a_0 + \textstyle\sum_{i=1}^{k}\gamma_i\left(\sum_{j=1}^{k}\gamma_j a_{i,j} + \sum_{j=k+1}^{d} a_{i,j}\right) \\ + \textstyle\sum_{i=k+1}^{d}\sum_{j=k+1}^{d} a_{i,j} + \sum_{i=k+1}^{d} a_{i,i}. \end{aligned}$$

From Proposition 3.5.12 it is straightforward to derive a recursive formula for the marginal proabilities which allows to sample from the additive linear family. For details see Procedure 10

*Proof.* We margin out the last component $d$. Let $I = [\![1, d-1]\!]$,

$$\begin{aligned} q^{(d-1)}_{\mathbf{A},a_0}(\boldsymbol{\gamma}_I)\nu^{-1} &= \left(q^{(d)}_{\mathbf{A},a_0}(\boldsymbol{\gamma}_I, 1) + q^{(d)}_{\mathbf{A},a_0}(\boldsymbol{\gamma}_I, 0)\right)\nu^{-1} \\ &= 2a_0 + (\boldsymbol{\gamma}_I, 1)^\intercal \mathbf{A}(\boldsymbol{\gamma}_I, 1) + (\boldsymbol{\gamma}_I, 0)^\intercal \mathbf{A}(\boldsymbol{\gamma}_I, 0) \\ &= 2a_0 + \operatorname{tr}\left(\mathbf{A}\left[(\boldsymbol{\gamma}_I, 1)(\boldsymbol{\gamma}_I, 1)^\intercal + (\boldsymbol{\gamma}_I, 0)(\boldsymbol{\gamma}_I, 0)^\intercal\right]\right) \\ &= 2a_0 + \operatorname{tr}\left(\mathbf{A}\begin{bmatrix} 2\boldsymbol{\gamma}_I\boldsymbol{\gamma}_I^\intercal & \boldsymbol{\gamma}_I \\ \boldsymbol{\gamma}_I^\intercal & 1 \end{bmatrix}\right) \end{aligned}$$

Iterating the argument, we obtain for $I = [\![1, d-t]\!]$ and $I^c := D \setminus I$

$$q^{(d-t)}_{\mathbf{A},a_0}(\boldsymbol{\gamma}_I)\nu^{-1} = 2^t a_0 + 2^{t-2}\operatorname{tr}\left(\mathbf{A}\begin{bmatrix} 4\,\boldsymbol{\gamma}_I\boldsymbol{\gamma}_I^\intercal & 2\,\boldsymbol{\gamma}_I\mathbf{1}_t^\intercal \\ 2\,\mathbf{1}_t\boldsymbol{\gamma}_I^\intercal & \mathbf{1}_t\mathbf{1}_t^\intercal + \mathbf{I}_t \end{bmatrix}\right)$$

Straightforward calculations yield

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

The proof is complete. □

Recall the remark on marginal distributions and moments we made in Section 3.2.1. For $\boldsymbol{\gamma}_I = \mathbf{1}$ we obtain

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

and $\pi_I(\mathbf{1}_k) = \nu 2^{d-|I|-2} s_I(\mathbf{1}_k)$ is indeed the expression for the cross-moments in the proof of Proposition 3.5.8.

**Algorithm 10:** Sampling from the additive linear family

$\boldsymbol{x} = (0,\dots,0),\ m \in (0,1),\ \mathbf{A} \in \mathbb{R}^{d\times d}$
$u \leftarrow U \sim \mathcal{U}_{[0,1]}$
**if** $u < m$ **then** $x_1 \leftarrow 1$, $\tilde{\mu} \leftarrow m$ **else** $x_1 \leftarrow 0$, $\tilde{\mu} \leftarrow 1 - m$
$p \leftarrow \tilde{\mu}$
**for** $i = 2$ **to** $d$ **do**
  $t \leftarrow 2^{d-(i+2)}(2\,|\boldsymbol{x}_{1:i-1}| + \sum_{j=i}^{d} a_{ij})$
  $\mu \leftarrow \tilde{\mu}/2 + t,\ c \leftarrow (\mu/\tilde{\mu} \vee 0) \wedge 1$
  $u \leftarrow U \sim \mathcal{U}_{[0,1]}$
  **if** $u < c$ **then**
    $x_i \leftarrow 1$
    $\tilde{\mu} \leftarrow \mu$
    **if** $c = 0$ **then** $p \leftarrow 0$ **else** $p \leftarrow pc$
  **else**
    $\tilde{\mu} \leftarrow \tilde{\mu} - \mu$
    **if** $c = 1$ **then** $p \leftarrow 0$ **else** $p \leftarrow p(1-c)$
  **end**
**end**
**return** $\boldsymbol{x},\ p$

### Method of moments

Given the cross-moments $\mathbf{M}$ with $\boldsymbol{m} = \operatorname{diag}(\mathbf{M})$, we can determine $a_0$ and a matrix $\mathbf{A}$ such that the family $q^a_{\mathbf{A},a_0}$ fits the desired cross-moments by solving a linear system of dimension $d(d+1)/2 + 1$. We first use the bijection

$$\tau\colon D \times D \to [\![1, d(d+1)/2]\!], \quad \tau(i,j) = i(i-1)/2 + j$$

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

$$\tilde{a}_{\tau(i,j)} := a_{ij},\ \tilde{m}_{\tau(i,j)} := m_{ij}, \quad i,j \in D$$

and the weight matrix

$$\tilde{s}_{\tau(i,j),\tau(k,l)} := 2^{\mathbb{1}_{\{i,j\}}(k)+\mathbb{1}_{\{i,j,k\}}(l)}, \quad i,j,k,l \in D.$$

Note that $|\tilde{\boldsymbol{a}}| = \mathbf{1}^\intercal\mathbf{A}\mathbf{1} + \operatorname{tr}(\mathbf{A})$. We then equate the distribution moments to the desired moments and normalize such that

$$2^{d-2}\left[\mathbf{I}\,a_0 + \frac{1}{4}\tilde{\mathbf{S}}\tilde{\boldsymbol{a}}\right] = \tilde{\boldsymbol{m}}, \quad 2^{d-2}(4a_0 + |\tilde{\boldsymbol{a}}|) = 1.$$

The solution of the linear system

$$\begin{pmatrix}\tilde{\boldsymbol{a}}^* \\ a_0^*\end{pmatrix} = 2^{-d+2}\begin{bmatrix}\frac{1}{4}\tilde{\mathbf{S}} & \mathbf{1} \\ 4\,\mathbf{1}^\intercal & 1\end{bmatrix}^{-1}\begin{pmatrix}\tilde{\boldsymbol{m}} \\ 1\end{pmatrix}$$

is finally transformed back into a symmetric matrix $\mathbf{A}^*$. The function $q^a_{\mathbf{A}^*,a_0^*}$ might not define a probability distribution, but for the average holds

$$\textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} \boldsymbol{\gamma}\boldsymbol{\gamma}^\intercal q^a_{\mathbf{A}^*,a_0^*}(\boldsymbol{\gamma}) = \mathbf{M}.$$

The weight matrix $\tilde{\mathbf{S}}$ does not depend on the data, and we could therefore fit the parameter to different cross-moment matrices on the same space $\mathbb{B}^d$ extremely fast once the weight matrix is build up in the memory.

## 3.6. Practical scope

In this section, we compare the $\mu$-conditionals family with logistic, linear and arctan link functions to the copula families with normal and student's t auxiliary distributions. We draw random cross-moment matrices of varying dimension and difficulty, fit the parametric families and record how well the desired correlation structure can be reproduced on average.

### 3.6.1. Sparse families

The major drawback of any kind of multiplicative model is the fact that we have no closed-form likelihood-maximizers, and therefore the parameter estimation requires

costly iterative fitting procedures. We can considerably speed up the parameter estimation if we work with a sparse version of the original parametric family which we might estimate a lot faster than the saturated family. For a proposal distribution, it is particularly important to take the strong dependencies into account but it is usually sufficient to work with very sparse families.

Instead of fitting the saturated model $q(\gamma_i \mid \gamma_{1:i-1})$, we preferably work with a more parsimonious regression model like $q(\gamma_i \mid \gamma_{C_i})$ for some index set $C_i \subseteq [\![1, i-1]\!]$, where the number of predictors $|C_i|$ is typically smaller than $i-1$. We solve this nested variable selection problem using some simple, fast to compute criterion.

Given the weighted data $\boldsymbol{w} \in [0,1]^n$, $\mathbf{X} \in \mathbb{B}^{n \times d}$, we denote the weighted sample cross-moments by $\overline{\mathbf{X}} = \sum_{k=1}^n w_k \boldsymbol{x}_k \boldsymbol{x}_k^\intercal$ and the weighted sample correlation by

$$r_{ij} = \frac{\overline{x}_{ij} - \overline{x}_{ii}\overline{x}_{jj}}{\sqrt{\overline{x}_{ii}(1-\overline{x}_{ii})\overline{x}_{jj}(1-\overline{x}_{jj})}}.$$

For $\varepsilon = 0.02$, we define the index set

$$I := \{ i \in D \mid \ \overline{x}_{ii} \ \notin \ (\varepsilon, 1-\varepsilon) \}.$$

which identifies the components which have, according to the data, a marginal probability close to either boundary of the unit interval. For the components $i \in I$, we do not model any dependencies but draw them independently of the other components. Dependencies do not really matter if the marginal probability is excessively small or large, but the components $i \in I$ are prone to cause complete separation in the data or might even be constant. For a $\mu$-conditionals family, we set $a_{ii} = \mu^{-1}(\overline{x}_{ii})$ and $\boldsymbol{a}_{i,-i} = \mathbf{0}$; for a copula family, we set all correlation coefficients in the target correlation matrix to zero.

For the remaining components $D \setminus I$, we construct sparse families in the sense that for $\delta \in (0,1)$, we define the index sets

$$C_i := \{ j \in [\![1, i-1]\!] \mid \delta < |r_{ij}| \}, \quad i \in D \setminus I,$$

which identify the components with index smaller than $i$ and significant mutual association. For a $\mu$-conditionals family, we model the conditional proababilities only with respect to the components in $C_i$ which means that $q(\gamma_i \mid \boldsymbol{\gamma}_{1:i-1}) = \mu(\sum_{j \in C_i} a_{ij}\gamma_j)$; for a copula family, we set the correlation coefficients $\sigma_{ij}$ in the target correlation matrix to zero for all $j \notin C_i$.

In the context of the SMC sampler, running algorithm on the examples in Section 4.5 with $\delta = 0$ and $\delta = 0.075$ reveals that a saturated logistic conditionals family achieves about the same acceptance rates as a sparse one, while the latter needs dramatically less computational time in the calibration step.

### 3.6.2. Random cross-moment matrices

We briefly discuss how to generate a valid random cross-moment matrix of binary data. We easily sample the mean $\boldsymbol{m} = \text{diag}(\mathbf{M}) \sim \mathcal{U}_{(0,1)^d}$, but for the off-diagonal elements we have to ensure that the covariance matrix $\mathbf{M} - \boldsymbol{m}\boldsymbol{m}^\intercal$ is positive definite and that the constraints (3.1) are all met. We alternate the following two steps.

- Permutations $m_{ij} = m_{\sigma(i)\sigma(j)}$ for $i, j \in D$ with uniform $\sigma \sim \mathcal{U}_{S(D)}$ where we denote by $S(D) := \{\sigma\colon D \to D, \sigma \text{ is bijective}\}$ the set of all permutations on $D$.
- Replacements $m_{id} = m_{di} \sim \mathcal{U}_{[a_i,b_i]}$ for all $i = \sigma(1), \ldots, \sigma(d-1)$ with uniform $\sigma \sim \mathcal{U}_{S(D\setminus\{d\})}$ where the bounds $a_i, b_i$ are subject to the constraints $\det(\mathbf{M}) > 0$ and $\min\{m_{ii} + m_{dd} - 1, 0\} \leq m_{id} \leq \max\{m_{ii}, m_{dd}\}$.

The replacement step needs some consideration. We denote by $\mathbf{N}$ the inverse of the $(d-1) \times (d-1)$ upper sub-matrix of $\mathbf{M}$ and define $\tau_i := m_{di} \sum_{i \in D\setminus\{d\}} m_{dj} n_{ij}$ such that $\det(\mathbf{M}) = [1/\det(\mathbf{N})](m_{dd} - \sum_{i \in D\setminus\{d\}} \tau_i)$. If we replace $m_{di} = m_{id}$ by $x_i$ we have to ensure that $\det[\mathbf{M}(x_i)] = \det(\mathbf{M}) + m_{di}(m_{di} n_{ii} + 2\tau_i) - x_i(x_i n_{ii} + 2\tau_i) > 0$ which means $(x_i + \tau_i/n_{ii}) \in (-c_i, c_i)$ with $c_i := [\tau_i^2/n_{ii}^2 + \det(\mathbf{M}) + m_{di}(m_{di} n_{ii} + 2\tau_i)]^{-1/2}$. Therefore, the lower and upper bounds, $a_i := \max\{m_{ii} + m_{dd} - 1, 0, -\tau_i/n_{ii} - c_i\}$ and $b_i := \min\{m_{ii}, m_{dd}, -\tau_i/n_{ii} + c_i\}$, respect all constraints on $x_i$. We rapidly update the value of the determinant $\det[\mathbf{M}(x_i)]$ and proceed with the next entry.

We perform $10 \cdot d$ permutation steps and run 500 sweeps of replacements between permutations. The result is approximately a uniform draw from the set of feasible cross-moments matrices. However, sampling according to these cross-moments might not be possible in higher dimensions because the cross-moment matrix is likely to contain extreme cases which are beyond the scope of the parametric family or not workable for numerical reasons. We introduce a parameter $\varrho \in [0, 1]$ which governs the difficulty of the sampling problem by shrinking the upper and lower bounds $a$ and $b$ of the uniform distributions to $a^\varrho := [(1+\varrho)a + (1-\varrho)b]/2$ and $b^\varrho := [(1-\varrho)a + (1+\varrho)b]/2$, respectively.

#### Sampling binary data with specified cross-moment matrix

If $2^d - 1$ full probabilities are known, we easily sample from the corresponding multinomial distribution (Walker, 1977). For a valid set of cross-moments $m_I$, $I \in \mathcal{I}$, Gange (1995) proposes to compute the full probabilities using a variant of the Iterative Proportional Fitting algorithm (Haberman, 1972). While there are no restrictions on the range of dependencies, we have to enumerate the entire state space which limits this versatile approach to low dimensions.

In the sequel, we do not consider methods for structured correlations nor approaches which require enumeration of the state space. First, we show how to compute the parameter $\mathbf{A}$ of a $\mu$-conditionals model for a given cross-moment matrix $\mathbf{M}$. Secondly, we review an alternative approach to sampling binary data with given cross-moment matrix $\mathbf{M}$ based on the copula of an underlying auxiliary parametric family.

## 3.6.3. Computational results

### Figure of merit

Let $\mathbf{M}$ be a cross-moments matrix and let $\mathbf{M}^*$ denote the cross-moment matrix with mean $\boldsymbol{m} = \operatorname{diag}(\mathbf{M})$ and uncorrelated entries $m^*_{ij} = m_{ii}m_{jj}$ for all $i \neq j \in D$. For a parametric family $q_\theta$, we define the figure of merit

$$\tau_q(\mathbf{M}) := (\|\mathbf{M} - \mathbf{M}^*\| - \|\mathbf{M} - \mathbf{M}^q\|)/\|\mathbf{M} - \mathbf{M}^*\|, \tag{3.5}$$

where $\mathbf{M}^q$ denotes the sampling cross-moment matrix of the parametric family with parameter $\theta$ adjusted to the desired cross-moment matrix $\mathbf{M}$. The norm $\|\cdot\|$ might be any non-trivial matrix norm; in our numerical experiments we use the spectral norm $\|\mathbf{A}\|_2^2 := \lambda_{\max}(\mathbf{A}^\intercal\mathbf{A})$, where $\lambda_{\max}$ delivers the largest eigenvalue, but we found the Frobenius norm $\|\mathbf{A}\|_F^2 := \operatorname{tr}(\mathbf{A}\mathbf{A}^\intercal)$ to provide qualitatively the very same picture.

We can roughly interpret $\tau_q(\mathbf{M})$ as the proportion of the correlation structure that the parametric family is able to reproduce. The score $\tau_q(\mathbf{M})$ is negative if the parametric family $q_\theta$ performs worse than $q^{\sqcap}_{\boldsymbol{m}}$.

### Setup

For fitting the logistic conditionals family when $d > 10$, we replace the exact terms by Monte Carlo estimates (3.2) where we use $n = 10^4$ random samples. We estimate the cross-moment matrix of the parametric family $q$ by $\mathbf{M}^q \approx n^{-1}\sum_{k=1}^n \boldsymbol{x}_k\boldsymbol{x}_k^\intercal$ where we use $n = 10^6$ samples from $q$. This concerns only the logistic and linear conditionals families; for the copula families, we can explicitly compute the sampling cross-moments as $m^q_{ij} = \Phi_2(\mu_i, \mu_j; \sigma_{ij})$, where $\boldsymbol{\Sigma}$ is the adjusted correlation matrix of the underlying multivariate normal distribution made feasible via (3.3).

We loop over 15 levels of difficulty $\varrho \in [0, 1]$ in 3 dimensions $d = 10, 25, 50$, and generate at each time 200 cross-moments matrices. We denote by $\tau_1 \leq \cdots \leq \tau_{200}$ the ordered figures of merit of the random cross-moment matrices. We report the median and

the quantiles $(\tau_{\lfloor(0.5-\omega)n\rfloor}, \tau_{\lceil(0.5+\omega)n\rceil})$, depicted as underlying gray areas for 20 equidistant values of $\omega \in [0.0, 0.5]$. Figures 1-3 show the results grouped by parametric families; the $y$-axis with the scale on the left represents the figure of merit $\tau \in [0, 1]$, the $x$-axis represents the level of difficulty $\varrho \in [0, 1]$, and the $[0.0, 0.5]$-gray-scale on the right refers to the level of the quantiles.

### 3.6.4. Discussion

While Theorem 3.3.2 states that a $\mu$-conditionals family can encompass any feasible mean and correlation structure, we cannot entirely turn this into practice. If the desired correlations are difficult to model, the limited numerical accuracy on a computer does not allow to exactly reproduce the correlation structure using a $\mu$-conditionals family. However, the scope of the copula methods presented in Section 3.4 is already limited by their mathematical structure.

The copula families are guaranteed to have the correct mean but they are less flexible than the conditionals families; besides, they do not allow for fast point-wise evaluation of their mass functions. The student's t family seems to outperform the normal family on moderately difficult instances, while the latter seems to work relatively better on difficult instances.

The truncated linear conditionals family is fast to compute but its quality deteriorates rapidly with growing complexity. The logistic and arctan conditionals families seem to perform equally well, the latter having slightly less outliers and betters scores on moderately difficult instances. They are computationally demanding but by far the most versatile option.

These findings confirm comparisons carried out against the backdrop of particular applications (Farrell and Rogers-Stewart, 2008; Schäfer and Chopin, 2012; Schäfer, 2012b), see Sections 4.4.2 and 6.4.1 for toy examples. In the following chapters on applications, we primarily use the logistic conditionals family as sampling distribution. The advantage of the logistic over the arctan link function is that the logistic link function yields concave likelihood-functions and component-wise likelihood-maximization can be performed using standard methods like Newton-Raphson.

**Figure 3.1.:** Logistic conditionals family

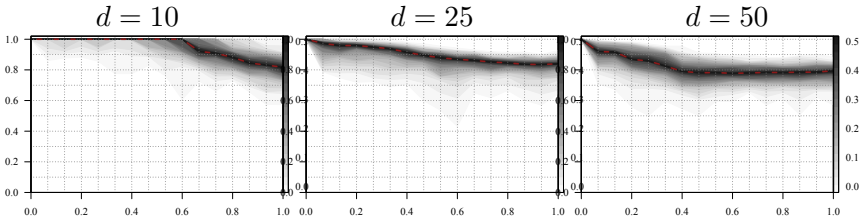



**Figure 3.2.:** Arctan conditionals family

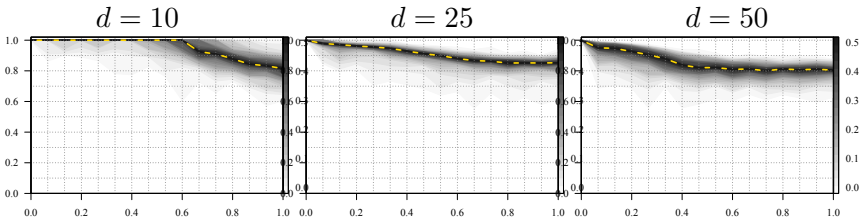



**Figure 3.3.:** Truncated linear conditionals family

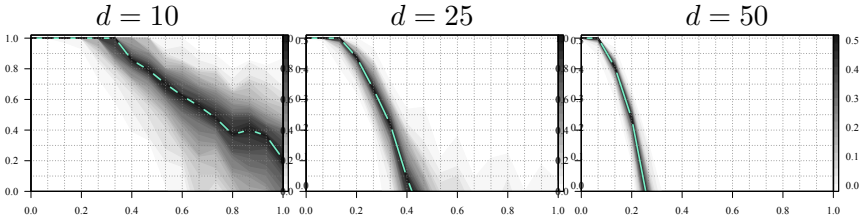



**Figure 3.4.:** Student's t copula family

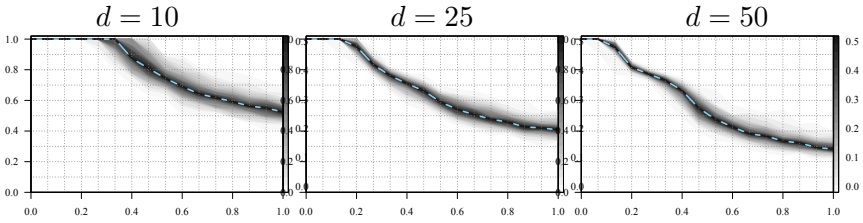
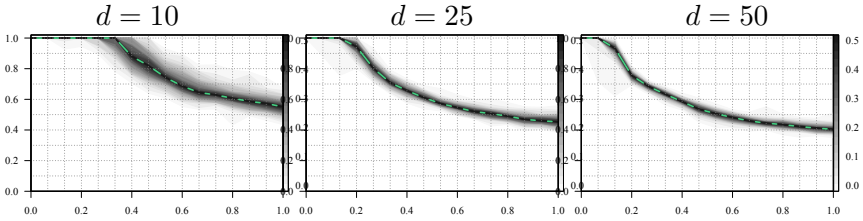



**Figure 3.5.:** Normal copula family



# Part II.

# Applications

# 4. Bayesian variable selection for normal linear models

## Resumé

*L'application statistique majeure pour d'échantillonnage de vecteurs binaires est la sélection bayésienne de variables parmi des modèles de régression linéaire où des quantités telles que les probabilités d'inclusion a posteriori des prédicteurs doivent être calculées. Ce chapitre propose une brève introduction à la sélection de variables dans le cadre de modèles linéaires normaux, où la distribution a posteriori est disponible sous forme analytique pour un choix judicieux de la loi a priori sur les paramètres du modèle. Nous construisons plusieurs instances de test exigeants sur données réelles, choisis pour être considérablement multimodal, et l'échantillonneur de Monte Carlo séquentiel est comparé avec des méthodes standards de Monte Carlo à chaîne de Markov (George and McCulloch, 1997).*

## 4.1. Introduction

We apply the sequential Monte Carlo (SMC) sampler developed in Chapter 2 to Bayesian variable selection in the context of normal linear models. The numerical examples are taken from Schäfer and Chopin (2012).

Let $Y$ denote the random quantity of interest or *response* and $\boldsymbol{Z}$ a $d$-dimensional vector of *covariates* or *predictors*. For real valued response variables, the generic choice is the *linear normal* model

$$h(y, \boldsymbol{z}) = [\sigma\sqrt{2\pi}]^{-1} \exp\left[-(y - \alpha - \boldsymbol{\beta}^{\intercal}\boldsymbol{z})^2/(2\sigma^2)\right]. \tag{4.1}$$

In the sequel, we write $h(y \mid \boldsymbol{z})$ instead of $h_{Y|\boldsymbol{Z}}(y \mid \boldsymbol{z})$ if the arguments of the conditional density or mass function unambiguously indicate which distribution we are referring to.

We denote by $n$ the number of observation, by $\boldsymbol{y} \in \mathbb{R}^n$ the vector of observed explained variables and by $\mathbf{Z} \in \mathbb{R}^{n \times d}$ the design matrix of observed explanatory variables. We always assume the observations to be independent, and the design matrix to be of full rank with columns centered such that $\mathbf{1}^\intercal \mathbf{Z} = \mathbf{0}$.

### 4.1.1. Selection criteria

In variable selection, the idea is to identify a subset of all available predictors which balances the explanatory power and the complexity of the model. In the regression context, it is convenient to identify each model with a binary vector $\boldsymbol{\gamma} \in \mathbb{B}^d$ where the predictor $Z_i$ is in the model if and only if $\gamma_i = 1$. Usually, a criterion of goodness-of-fit

$$\tilde{\pi}(\cdot \mid \boldsymbol{y}, \mathbf{Z}) \colon \mathbb{B}^d \to [0, \infty)$$

is defined which allows to rank the models based on the observed data. These functions rarely have any particular structure and tend to be quite multi-modal depending on the correlation between the predictors. The normalized criterion $\pi \propto \tilde{\pi}$ is a probability distribution, and Monte Carlo methods like Markov chain Monte Carlo (MCMC) discussed in Section 1.2 can provide an estimate of

$$\pi(f \mid \boldsymbol{y}, \mathbf{Z}) = \textstyle\sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} f(\boldsymbol{\gamma})\, \pi(\boldsymbol{\gamma} \mid \boldsymbol{y}, \mathbf{Z}), \tag{4.2}$$

where $f$ might be any quantity of interest. The most important examples are probably $f(\boldsymbol{\gamma}) = \boldsymbol{\beta}_{\boldsymbol{\gamma}}$ for the average regression coefficients and $f(\boldsymbol{\gamma}) = \boldsymbol{\gamma}$ for the average inclusion of the predictors. In a Bayesian context, $\pi(\cdot \mid \boldsymbol{y}, \mathbf{Z})$ has an interpretation as the posterior probability distribution and concepts like Bayesian model averaging (Hoeting et al., 1999) or the median model (Barbieri and Berger, 2004) depend on methods which can reliably estimate (4.2).

The convergence rates of MCMC based approaches slow down dramatically as the dimension $d$ grows and the multimodality of the target distribution increases. This motivates the use of the SMC sampler described in Chapter 2 which we show to largely outperform standard MCMC algorithms on difficult instances of Bayesian variable selection in linear normal models (4.1) with about 100 predictors, see Section 4.5. We exploit the fact that the SMC sampler allows for straightforward parallelization and provide examples with 1500 predictors to underpin its potential for solving high-dimensional problems in parallel computing environments.

### 4.1.2. Bayesian variable selection

Later, we mainly concentrate on Bayesian variable selection approaches where the derived criterion is the a posteriori distributions on the model space. We denote the likelihood given the model by

$$\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \alpha, \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \theta, \boldsymbol{\gamma}) := \prod_{k=1}^{n} h(y_k, \boldsymbol{z}_k \mid \boldsymbol{\gamma}, \alpha, \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \theta),$$

where $\alpha$ and $\boldsymbol{\beta}$ are the regression coefficients and $\theta$ denotes further nuisance parameters. For a suitable prior distribution $p$ on these nuisance parameters, the marginal likelihood

$$\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma}) = \int \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \alpha, \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \theta, \boldsymbol{\gamma}) p(\alpha, \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \theta \mid \boldsymbol{\gamma}) d(\alpha, \boldsymbol{\beta}, \theta)$$

can be computed and via Bayes' Theorem

$$\pi(\boldsymbol{\gamma} \mid \boldsymbol{y}, \mathbf{Z}) \propto \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma}) p(\boldsymbol{\gamma})$$

one obtains an unnormalized version of the posterior distribution on the model space.

### 4.1.3. Penalized likelihood criteria

In a more Frequentist framework, one might rank the models according to some penalized likelihood criterion. We briefly review two popular approaches for model selection.

The Bayesian information criterion (BIC) was first proposed by Schwarz (1978) and can be derived as the logarithm of a second degree Laplace approximation to the marginal likelihood (4.1.2),

$$\text{BIC}(\boldsymbol{\gamma}) := \log \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma}, \hat{\alpha}, \hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \hat{\theta}) - \frac{|\boldsymbol{\gamma}|}{2} \log n \simeq \log \pi(\boldsymbol{\gamma} \mid \boldsymbol{y}, \mathbf{Z}),$$

where $\hat{\alpha}, \hat{\boldsymbol{\beta}}, \hat{\theta}$ are the maximum-likelihood estimates of the nuisance parameters and $n$ the number of observations. The symbol $\simeq$ means approximation up to an additive constant. Asymptotically, the BIC coincides with the Bayesian approach for certain choices of the prior distributions.

The so-called Akaike information criterion (AIC) developed by Akaike (1974) is based on information theoretic reasoning and penalizes the complexity independently of the number of observations,

$$\text{AIC}(\boldsymbol{\gamma}) := \log \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma}, \hat{\alpha}, \hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \hat{\theta}) - |\boldsymbol{\gamma}| \,.$$

The AIC can be shown to asymptotically minimize the information loss in terms of Kullback–Leibler divergence. There are also correction for finite sample sizes.

### 4.1.4. Convex optimization

For linear normal models, an alternative to likelihood-based selection criteria are regularized versions of least squares estimates,

$$\boldsymbol{\beta}^* = \operatorname{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^d} \left[\|\boldsymbol{y} - \mathbf{Z}\boldsymbol{\beta}\|_2^2 + p(\boldsymbol{\beta})\right],$$

where $p > 0$ is a penalty term. For certain continuous but non-smooth penalty functions the coefficients are shrunk to zero which is a model selection procedure.

The *least absolute shrinkage and selection operator* (Tibshirani, 1996, LASSO) with penalty function $p(\boldsymbol{\beta}) = \theta\,|\boldsymbol{\beta}|$ is probably the most prominent example of regularized least squares for model selection. The minimization problem can be solved using convex optimization techniques which allow to solve problems which are too large to be efficiently treated using likelihood-based criteria. There are various variants and extensions like the *least angle regression* (Efron et al., 2004, LARS), the *elastic net* (Zou and Hastie, 2005) and the *smoothly clipped absolute deviation* (Fan and Li, 2001, SCAD) algorithms which have been subject to intensive research in the recent years. See Celeux et al. (2011) for a comparison of regularization techniques and Bayesian approaches.

## 4.2. Marginal likelihood

In this section, we review strategies to assigning prior distributions to the parameters of the linear normal model which allow to obtain a closed-form expression for the marginal likelihood where all parameters except for the model indicator are integrated out. The linear normal model has the full likelihood

$$\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \alpha, \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \sigma^2, \boldsymbol{\gamma}) \propto \sigma^{-n} \exp\left[-\frac{1}{2\sigma^2}[(\alpha\mathbf{1} + \mathbf{Z}_{\boldsymbol{\gamma}}\boldsymbol{\beta}_{\boldsymbol{\gamma}} - \boldsymbol{y})^\intercal(\alpha\mathbf{1} + \mathbf{Z}_{\boldsymbol{\gamma}}\boldsymbol{\beta}_{\boldsymbol{\gamma}} - \boldsymbol{y})]\right],$$

where the intercept $\alpha$ does not depend on the model since we assume the design matrix to be centered. For an improper prior $p(\alpha) \propto 1$, the marginal likelihood becomes

$$\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \sigma^2, \boldsymbol{\gamma}) \propto \sigma^{-(n-1)} \exp\left[-\frac{1}{2\sigma^2}[(\overline{\boldsymbol{y}} + \mathbf{Z}_{\boldsymbol{\gamma}}\boldsymbol{\beta}_{\boldsymbol{\gamma}} - \boldsymbol{y})^\intercal(\overline{\boldsymbol{y}} + \mathbf{Z}_{\boldsymbol{\gamma}}\boldsymbol{\beta}_{\boldsymbol{\gamma}} - \boldsymbol{y})]\right],$$

that is $\alpha = \overline{y}$ is just the least squares estimate where $\overline{y} := n^{-1}\boldsymbol{y}^\intercal\mathbf{1}$ and $\overline{\boldsymbol{y}} := \overline{y}\mathbf{1}$. For each model, we define the orthogonal projection

$$\Pi_{\boldsymbol{\gamma}}^{\perp}\colon \mathbb{R}^n \to \{\mathbf{Z}_{\boldsymbol{\gamma}}\boldsymbol{\beta}_{\boldsymbol{\gamma}} \mid \boldsymbol{\beta}_{\boldsymbol{\gamma}} \in \mathbb{R}^{|\boldsymbol{\gamma}|}\} \subset \mathbb{R}^n,\ \Pi_{\boldsymbol{\gamma}}^{\perp} := \mathbf{Z}_{\boldsymbol{\gamma}}(\mathbf{Z}_{\boldsymbol{\gamma}}^\intercal\mathbf{Z}_{\boldsymbol{\gamma}})^{-1}\mathbf{Z}_{\boldsymbol{\gamma}}^\intercal.$$

The residual, explained and total sum of squares are related through Pythagoras' Theorem $\|\overline{\boldsymbol{y}} + \Pi_{\boldsymbol{\gamma}}^{\perp}\boldsymbol{y} - \boldsymbol{y}\|_2^2 + \|\Pi_{\boldsymbol{\gamma}}^{\perp}\boldsymbol{y}\|_2^2 = \|\boldsymbol{y} - \overline{\boldsymbol{y}}\|_2^2$. The coefficient of determination is defined by $R_{\boldsymbol{\gamma}}^2 := \|\Pi_{\boldsymbol{\gamma}}^{\perp}\boldsymbol{y}\|_2^2/\|\boldsymbol{y} - \overline{\boldsymbol{y}}\|_2^2$.

### 4.2.1. Hierarchical priors

For a judicious choice of prior distributions, there are analytic expressions for the marginal likelihood (4.1.2) which allows to evaluate the posterior distribution $\pi$ of each model up to a constant. Then the prior takes the form

$$p(\boldsymbol{\beta}_{\boldsymbol{\gamma}}, \sigma^2 \mid \boldsymbol{\gamma}) = p(\boldsymbol{\beta}_{\boldsymbol{\gamma}} \mid \sigma^2, \boldsymbol{\gamma}) p(\sigma^2 \mid \boldsymbol{\gamma})$$

where the prior on $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ is multivariate normal $p(\cdot \mid \sigma^2, \boldsymbol{\gamma}) = \mathcal{N}(\mathbf{0}, \sigma^2 \tau \boldsymbol{\Sigma}_{\boldsymbol{\gamma}})$ with dispersion parameter $\tau > 0$ and positive matrix $\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}$, and the prior on the residual variance $\sigma^2$ is inverse-gamma $p(\cdot \mid \boldsymbol{\gamma}) = \mathcal{I}(a/2, ab/2)$ with $a, b \geq 0$.

The typical choice for the covariance is either the identity matrix $\boldsymbol{\Sigma}_{\boldsymbol{\gamma}} = \mathbf{I}_{\boldsymbol{\gamma}}$ where we assume the correlation coefficients to be a priori independent, or the observed Fisher information matrix $\boldsymbol{\Sigma}_{\boldsymbol{\gamma}} = (\mathbf{Z}_{\boldsymbol{\gamma}}^\intercal \mathbf{Z}_{\boldsymbol{\gamma}})^{-1}$. The marginal likelihood is

$$\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma}) \propto \left| \boldsymbol{\Sigma}_{\boldsymbol{\gamma}} \mathbf{Z}_{\boldsymbol{\gamma}}^\intercal \mathbf{Z}_{\boldsymbol{\gamma}} + \tau^{-1} \mathbf{I}_{\boldsymbol{\gamma}} \right|^{-1/2} \left[ ab + \|\boldsymbol{y} - \overline{\boldsymbol{y}}\|_2^2 - \boldsymbol{y} \Pi_{\boldsymbol{\gamma}} \boldsymbol{y} \right]^{-(n-1+a)/2}.$$

where $\Pi_{\boldsymbol{\gamma}} = \mathbf{Z}(\mathbf{Z}^\intercal \mathbf{Z} + \tau^{-1} \boldsymbol{\Sigma}_{\boldsymbol{\gamma}}^{-1})^{-1} \mathbf{Z}^\intercal$ denotes the projection under the prior.

### 4.2.2. Zellner's prior

It is straightforward to see that the choice $\boldsymbol{\Sigma}_{\boldsymbol{\gamma}} = (\mathbf{Z}_{\boldsymbol{\gamma}}^\intercal \mathbf{Z}_{\boldsymbol{\gamma}})^{-1}$ has a computational advantage and an interesting interpretation. The projection under the prior is the scaled orthogonal projection $\Pi_{\boldsymbol{\gamma}} = s \Pi_{\boldsymbol{\gamma}}^\perp$ and the determinant is $s^{-|\boldsymbol{\gamma}|/2}$ where $s = \tau/(1+\tau)$ denotes the shrinkage factor. Further, for $a = b = 0$ we observe that

$$\|\boldsymbol{y} - \overline{\boldsymbol{y}}\|_2^2 - s \boldsymbol{y} \Pi_{\boldsymbol{\gamma}}^\perp \boldsymbol{y} = \|\boldsymbol{y} - \overline{\boldsymbol{y}}\|_2^2 + s \|\Pi_{\boldsymbol{\gamma}}^\perp \boldsymbol{y}\|_2^2 \propto 1 - s R_{\boldsymbol{\gamma}}^2,$$

allowing to express the marginal likelihood in terms of the coefficient of determination

$$\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tau, \boldsymbol{\gamma}) \propto (1+\tau)^{(n-1-|\boldsymbol{\gamma}|)/2} [1 + \tau(1 - R_{\boldsymbol{\gamma}}^2)^{-(n-1)/2}].$$

The choice for the dispersion parameter may be $\tau = n$ in reason of the unit information prior (Kass and Wasserman, 1996), $\tau = d^2$ based on the risk inflation criterion (Foster and George, 1994) or $\tau = \operatorname{argmax}_\tau \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tau, \boldsymbol{\gamma})$ for a local empirical prior (Hansen and Yu, 2001). We refer to Liang et al. (2008) for a more thorough discussion.

Some authors advocate to put a suitable prior on the dispersion parameter which provides thicker tails in the prior distribution and ensures that the posterior probabilities are consistent (Zellner and Siow, 1980; Liang et al., 2008). The generic choice might

be the inverse gamma prior $\mathcal{I}(a/2, ab/2)$ where the hyper parameters $a = 1$ and $b = n$ provide exactly a multivariate Cauchy prior on the regression parameters $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$. For the inverse gamma prior, the marginal likelihood

$$\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma}) = \int \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tau, \boldsymbol{\gamma}) p(\tau) d\tau$$

can be computed via numerical integration or by means of a Laplace approximation. The latter is particularly fast to compute since there is an analytic expression for the maximum of the integrand $\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tau, \boldsymbol{\gamma}) p(\tau)$. Liang et al. (2008) propose an alternative hyper prior $\pi(\tau) = (a-2)(1+\tau)^{-a/2}/2$ with $a > 2$ which allows to express the marginal likelihood in terms of certain Gaussian hyper geometric functions.

### 4.2.3. Independent prior

The independent prior is computationally less convenient. We might define the product $\boldsymbol{b}_{\boldsymbol{\gamma}} = \mathbf{Z}_{\boldsymbol{\gamma}}^{\intercal} \boldsymbol{y}$ and the Cholesky decomposition $\mathbf{C}_{\boldsymbol{\gamma},\tau} \mathbf{C}_{\boldsymbol{\gamma},\tau}^{\intercal} = \boldsymbol{\Sigma}_{\boldsymbol{\gamma}} \mathbf{Z}_{\boldsymbol{\gamma}}^{\intercal} \mathbf{Z}_{\boldsymbol{\gamma}} + \tau^{-1} \mathbf{I}_{\boldsymbol{\gamma}}$ which allows to write the posterior mass function as

$$\pi(\boldsymbol{\gamma} \mid \boldsymbol{y}, \mathbf{Z}) \propto \tau^{|\boldsymbol{\gamma}|/2} \prod_{i=1}^{|\boldsymbol{\gamma}|} c_{i,i}^{(\boldsymbol{\gamma},\tau)} \left[ ab + \|\boldsymbol{y} - \overline{\boldsymbol{y}}\|_2^2 - (\mathbf{C}_{\boldsymbol{\gamma},\tau}^{-1} \boldsymbol{b}_{\boldsymbol{\gamma}})^{\intercal} \mathbf{C}_{\boldsymbol{\gamma},\tau}^{-1} \boldsymbol{b}_{\boldsymbol{\gamma}} \right]^{-(n-1+a)}.$$

If one wants a full Bayesian approach having a prior on the dispersion parameter, Zellner's prior is to be preferred for its computational efficiency, since for the independent prior we cannot easily integrate out the dispersion parameter.

## 4.3. Priors on the model space

Typically, the prior distribution on the model space is

$$p(\boldsymbol{\gamma} \mid m) = m^{|\boldsymbol{\gamma}|} (1-m)^{d-|\boldsymbol{\gamma}|}$$

for some common prior marginal inclusion probability $m \in (0, 1)$. Some authors, e.g. Nott and Kohn (2005), propose a conjugate Beta hyper prior $m \sim B(a, b)$ for $a, b > 0$ which yields $p(\boldsymbol{\gamma}) = B(a+|\boldsymbol{\gamma}|-1, b+d-|\boldsymbol{\gamma}|)/B(a, b)$ where $B$ denotes the Beta function.

### 4.3.1. Prior on the model size

We propose to choose a uniform prior conditional on the size of the model and a binomial hyper prior $k \sim \mathcal{B}(m, d^*)$ on the size of the model which yields

$$p(\boldsymbol{\gamma}) = \sum_{k=0}^{d^*} \frac{k!(d^*-k)!}{d^*!} m^k (1-m)^{d^*-k} \delta_k(|\boldsymbol{\gamma}|) \frac{k!(d-k)!}{d!},$$

where $d^* \leq d \wedge n$ is the size of the largest admissible model. If $d > n$ one typically restricts the analysis to models of size $d$. Generally, if the number of predictors is large it usually suffices to only consider rather small models. The parameter $m = \overline{d}/d^*$ is chosen to yield a desired average model size $\overline{d} < d^*$.

### 4.3.2. Main effect restrictions

In some statistical applications, we add interactions between two predictors by crossing columns of the design matrix. The variable selection procedure remains the same, but typically the interaction should only be included in the model if the corresponding main effects are also present. For simplicity, we just consider two-way interactions and denote the interaction variables by $\tilde{\gamma}_{ij}$. For a variable selection problem of dimension $d(d-1)/2$, we define the prior

$$p(\boldsymbol{\gamma}) = \prod_{i,j \in D} m_i^{\gamma_i} (1 - m_i)^{1-\gamma_i} \, \tilde{m}_{ij}^{\tilde{\gamma}_{ij}\gamma_i\gamma_j} (1 - \tilde{m}_{ij})^{1-\tilde{\gamma}_{ij}},$$

where $\tilde{m}_{ij} = m_i m_j m_{ij}/(1 - m_{ij} + m_i m_j m_{ij})$ and $m_{ij} = \mathbb{P}(\gamma_{ij} = 1 \mid \gamma_i = 1, \gamma_j = 1)$. In particular, if $m_i = m_{ij} = 1/2$ for all $i, j \in D$, the prior is the uniform distribution on the constrained support $\{\boldsymbol{\gamma} \in \mathbb{B}^{d(d+1)/2} \mid \gamma_{ij} \leq \gamma_i\gamma_j, \ i, j \in D\}$. In the numerical experiments in Section 4.5, we show that adding these constraints makes the sampling problem even more challenging.

## 4.4. Sequential Monte Carlo

In this section, we provide some remarks on the sequence of intermediate distributions and the choice of the parametric families in the transition kernel against the backdrop of Bayesian variable selection.

### 4.4.1. Intermediate distributions

The SMC sampler as described in Chapter 2 uses a geometric bridge (2.2) to construct the sequence of intermediate distributions. However, there are other natural possibilities to obtain an auxiliary sequence of distribution in the context of variable selection.

**Data partition** Chopin (2002) proposes a static SMC sampler based on a sequence of posterior distributions where data is added as $\varrho_t$ increases. The auxiliary sequence is

$$\pi_t(\boldsymbol{\gamma}) = \pi(\boldsymbol{\gamma} \mid \boldsymbol{y}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{\lfloor \varrho_t n \rfloor}),$$

where $n$ is the total number of observations. The initial distribution $\pi_0(\boldsymbol{\gamma}) = \pi(\boldsymbol{\gamma} \mid \boldsymbol{y})$ is the prior $p(\boldsymbol{\gamma})$ on the model space. Note that for this scheme we cannot completely control the step size which makes it more difficult to calibrate the algorithm.

**Data orthogonalization** Ghosh and Clyde (2011) propose an orthogonal data augmentation scheme in the context of Gibbs sampling which could be incorporated into an SMC sampler. We can augment the data such that the design matrix $\mathbf{Z}_o = (\mathbf{Z}^\intercal, \mathbf{Z}_a^\intercal)^\intercal$ has orthogonal columns, where $\mathbf{Z}_a$ denotes the extra rows of the design matrix. We let $\boldsymbol{y}_o = (\boldsymbol{y}^\intercal, \boldsymbol{y}_a^\intercal)^\intercal$ where the pseudo-observations $\boldsymbol{y}_a$ are drawn from the full model. This setup leads to a sequence of posterior distributions based on a weighted sample

$$\pi_t(\boldsymbol{\gamma}) = \pi(\boldsymbol{\gamma} \mid \boldsymbol{y}, \mathbf{Z}, (1 - \varrho_t)\boldsymbol{y}_a, (1 - \varrho_t)\mathbf{Z}_a),$$

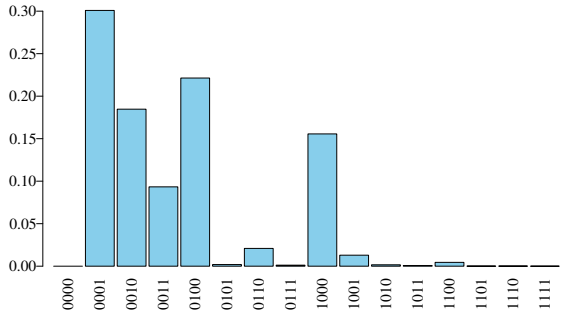
and for a uniform prior $p(\boldsymbol{\gamma}) = 2^{-d}$ on the model space, we have an initial distribution $\pi_0(\boldsymbol{\gamma}) = \pi(\boldsymbol{\gamma} \mid \boldsymbol{y}_o, \mathbf{Z}_o)$ with independent components. We could calibrate an optimal step size for this sequence but obviously the bi-sectional search would be more involved since each computation of the effective sample size in (2.5) requires evaluation of the target function $\pi_{\varrho_t + \alpha}(\boldsymbol{\gamma})$ for all particles.

**Geometric bridge** In our numerical studies, we stay with the geometric bridge (2.2) for its computational simplicity which allows to perfectly control the step size of the algorithm. Using the geometric bridge, we can start from any initial distribution $p$ with $\operatorname{supp}(\pi) \subseteq \operatorname{supp}(p)$ which allows to sample from $p$ and evaluate its mass function up to a constant. Intuitively, the SMC sampler converges faster if we choose an initial distribution which is, in a certain sense, closer to the distribution of interest. However, numerical experiments taught us that premature adjustment of $p$, for example using MCMC pilot runs, leads to faster but less robust algorithms. For Bayesian variable selection, we recommend to use the prior on the model space, see Section 4.3, as initial distribution which seems the natural choice in this context.

### 4.4.2. Parametric families

We briefly motivate why we need a parametric family which can model dependencies in order to make the Metropolis-Hastings independence sampler work in practice.

**Figure 4.1.:** True posterior mass function.




Suppose we have a simple linear relation $\boldsymbol{Y} = \boldsymbol{V}_1 + \boldsymbol{V}_2$. For $n = 100$ and $\mu = 10$, we draw a sample of normal distributed random variables

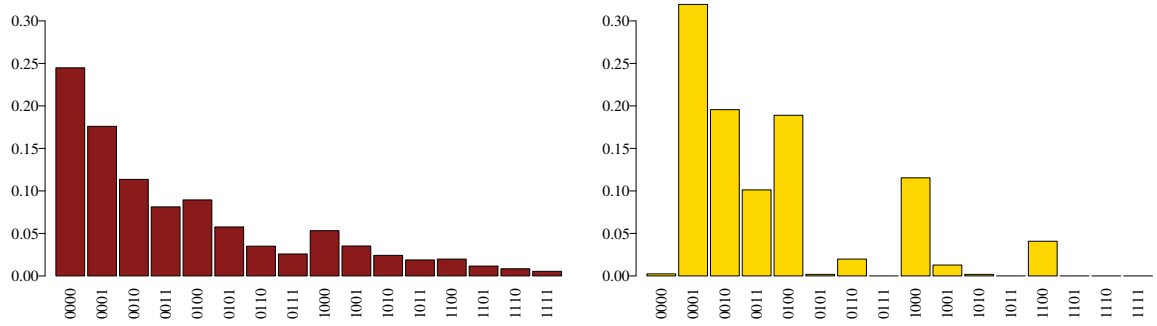
$$\boldsymbol{v}_1 \sim \mathcal{N}(-\mu, \mathbf{I}_n), \quad \boldsymbol{v}_2 \sim \mathcal{N}(\mu, \mathbf{I}_n),$$

and construct a vector of random observations $\boldsymbol{y} = \boldsymbol{v}_1 + \boldsymbol{v}_2$. Further, we draw four columns of predictors,

$$\boldsymbol{z}_1, \boldsymbol{z}_2 \sim \mathcal{N}\left(\boldsymbol{v}_1, (\mu^2/4)\,\mathbf{I}_n\right), \boldsymbol{z}_3, \boldsymbol{z}_4 \sim \mathcal{N}\left(\boldsymbol{v}_2, (\mu^2/4)\,\mathbf{I}_n\right).$$

The posterior distribution $\pi(\boldsymbol{\gamma}) = \pi(\boldsymbol{\gamma} \mid \boldsymbol{y}, \mathbf{Z})$, using the prior distributions as described in Section 4.2.1, typically exhibits strong dependencies between its components due to the correlation in the data.

**Figure 4.2.:** Approximations to the true posterior in Figure 4.1 by parametric families.



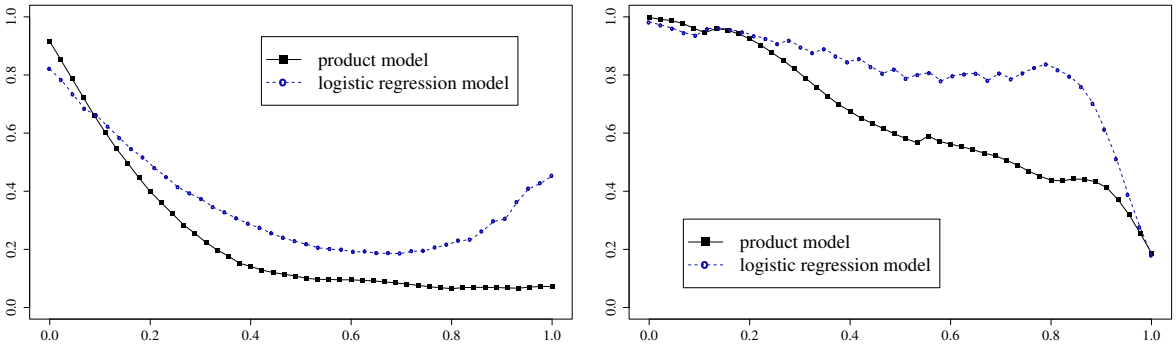

(a) product family $q_{\boldsymbol{m}}$ (b) logistic conditionals family $q_{\mathbf{A}}$

We now generate pseudo-random data $\mathbf{X}$ from $\pi$ and fit both a product family $q^{\square}_{\boldsymbol{m}}$ and a logistic conditionals family $q^{\ell}_{\mathbf{A}}$. Looking at the corresponding mass function in Figure 4.2, we notice how badly the product family mimics the true posterior. This observation carries over to larger sampling spaces.

An interesting way to further analyze the importance of reproducing the dependencies of $\pi$ is in terms of acceptance rates and particle diversities. The particle diversity defined in (2.6) naturally diminishes as our particle system approaches a strongly concentrated target distribution $\pi$. However, we want the SMC algorithm to keep the particle diversity up as long as possible to ensure that the particle system is well spread out over the entire state space of interest.

In Figure 4.3, we show a comparison (based on the Boston Housing data set explained in Section 4.5.1) between two SMC algorithms, using a product family and a

**Figure 4.3.:** We compare the use of a product family to a logistic conditionals family as proposal distribution of the Metropolis-Hastings kernel (1.15). We monitor a typical run ($\varrho$ on the x-axis) of our sequential Monte Carlo algorithm and plot the acceptance rates and particle diversities (on the y-axis).




(a) acceptance rates (b) particle diversities

logistic conditionals family as proposal distributions of the Metropolis-Hastings kernel (1.15). Clearly, in Figure 4.4(a), the acceptance rates achieved by the product kernel rapidly decrease and dwell around 5% for the second half of the run. In contrast, the logistic conditionals kernel always provides acceptance rates greater than 20%. As a consequence, in Figure 4.4(b), the particle diversity sustained by the product kernel decreases at an early stage, while the logistic regression kernel holds it up until the very last steps.

At first sight, it might seem odd that the acceptance rates of the logistic conditionals kernel increase during the final steps of the algorithm. If we jump ahead, however, and take a look at the results of the Boston Housing problem, see Figure 4.5(a), we notice that quite a few marginal probabilities of the posterior $\pi$ turn out to be zero, which makes it easier to reproduce the distributions towards the end of the resample-move algorithm. However, if we already decide at an early stage that a predictor has marginal probability zero, we fail to ever consider models containing this predictor for the rest of the algorithm. Therefore, the advantage of the logistic conditionals kernel over the simple product kernel is that we do not completely drop any components from the variable selection problem until the final steps.

## 4.5. Numerical experiments

For our numerical examples, we assume the regression parameters to be a priori independent, that is $\mathbf{\Sigma}_\gamma = \mathbf{I}_{|\gamma|}$. We follow the recommendations of George and McCulloch

(1997) and use the hyper-parameters

$$a = 4.0, \quad b = \widehat{\sigma}^2_{\mathbf{1}}, \quad \tau = 10.0/b \tag{4.3}$$

for the inverse gamma prior on the residual variance, where $\widehat{\sigma}^2_{\mathbf{1}}$ is the least square estimate of $\sigma^2$ based on the saturated model. The rationale behind this choice is to ensure a flat prior on the regression parameters $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ and to provide $\sigma^2$ with sufficient mass on the interval $(\hat{\sigma}^2_{\mathbf{1}}, \hat{\sigma}^2_{\mathbf{0}})$, where $\hat{\sigma}^2_{\mathbf{0}}$ denotes the variance of $\boldsymbol{y}$.

In this section we compare our SMC algorithm to standard MCMC methods based on local moves as introduced in Section 1.2. These are standard algorithms and widely used. There are other recent approaches like Bayesian adaptive sampling (Clyde et al., 2011) or evolutionary stochastic search (Bottolo and Richardson, 2010) which also aim at overcoming the difficulties of multi-modal binary distributions. However, a thorough and just comparison of our SMC approach to all other advanced methods is beyond the scope of this thesis.

### 4.5.1. Construction of test instances

For testing, we created variable selection problems with high dependencies between the covariates which yield particularly challenging, multi-modal posterior mass functions. The problems are built from freely available datasets by adding logarithms, polynomials and interaction terms. The MCMC methods presented in Section 1.2 tend to fail on these problems due to the very strong multi-modality of the posterior distribution while the SMC approach we advocate in Chapter 2 yields very reliable results. In the following, we briefly describe the variable selection problems composed for our numerical experiments.

**Boston Housing** The first example is based on the Boston Housing data set, originally treated by Harrison and Rubinfeld (1978), which is freely available at the StatLib data archive. The data set provides covariates ranging from the nitrogen oxide concentration to the per capita crime rate to explain the median prices of owner-occupied homes. The data has already been treated by several authors, mainly because it provides a rich mixture of continuous and discrete variables, resulting in an interesting variable selection problem. Specifically, we aim at explaining the logarithm of the corrected median values of owner-occupied housing. We enhance the 13 columns of the original data set by adding first order interactions between all covariates. Further, we add a constant column and a squared version of each covariate (except for CHAS since it is binary). This gives us a model choice problem with 104 possible predictors and

506 observations. By construction, there are strong dependencies between the possible predictors which leads to a rather complex, multi-modal posterior distribution.

| **short name** | **explanation** |
|---|---|
| CRIM | per capita crime |
| ZN | proportions of residential land zoned for lots over 2323 $m^2$ |
| INDUS | proportions of non-retail business acres |
| CHAS | tract borders Charles River (binary) |
| NOX | nitric oxides concentration (parts per $10^7$) |
| RM | average numbers of rooms per dwelling |
| AGE | proportions of owner-occupied units built prior to 1940 |
| DIS | weighted distances to five Boston employment centres |
| RAD | accessibility to radial highways |
| TAX | full-value property-tax rate per USD $10^4$ |
| PTRATIO | pupil-teacher ratios |
| B | $(Bk - 0.63)^2$ where Bk is the proportion of the black population |
| LSTAT | percentage of lower status population |

**Concrete Compressive Strength** The second example is constructed from a less known data set, originally treated by Yeh (1998), which is freely available at the UCI Machine Learning Repository. The data provides information about components of concrete to explain its compressive strength. The compressive strength appears to be a highly non-linear function of age and ingredients. In order to explain the compressive strength, we take the 8 covariates of the original data set and add the logarithms of some covariates (indicated by the prefix LG). Further, we add interactions between all 13 covariates of the augmented data set and a constant column. This gives us a model choice problem with 79 possible predictors and 1030 observations.

| **short name** | **explanation** |
|---|---|
| C, LG_C | cement |
| BLAST | blast furnace slag |
| FASH | fly ash |

| | |
|---|---|
| W, LG_W | water |
| PLAST | superplasticizer |
| CA, LG_CA | coarse aggregate |
| FA, LG_FA | fine aggregate |
| AGE, LG_AGE | age in days |

**Protein activity data** The third example has originally been analyzed by Clyde and Parmigiani (1998). Later, Clyde et al. (2011) used it as a challenging example problem in variable selection and included the raw data in the R-package BAS available at CRAN which implements the Bayesian Adaptive Sampling algorithm. In order to explain the protein activity (PROT.ACT1), we first convert the factors BUF, RA and DET into a factor model. We enhance the 14 columns of this data set by adding first order interactions between all covariates and a constant column. Note that some of the crossed columns turn out to be constant zeros such that we obtain a model choice problem with 88 possible predictors and 96 observations. For reasons of consistency, we choose the priors explained in the above Section 4.2.1 instead of Zellner's prior used by Clyde et al. (2011).

| **short name** | **explanation** |
|---|---|
| DET | detergent |
| BUF | pH buffer |
| NACL | salt |
| CON | protein concentration |
| RA | reducing agent |
| MGCL2 | magnesium chloride |
| TEMP | temperature |

## 4.5.2. Comparison and conclusion

We do not think it is reasonable to compare two completely different algorithms in terms of pure computational time. We cannot guarantee that our implementations are optimal nor that the time measurements can exactly be reproduced in other computing environments. We suppose that the number of evaluations of the target function $\pi$ is more of a fair stopping criterion, since it shows how well the algorithms exploit the information obtained from $\pi$. Precisely, we parameterize the SMC algorithm to not

exceed a fixed number $\nu$ of evaluations and stop the Markov chains when $\nu$ evaluations have been performed.

We compare the SMC sampler to both the adaptive Markov chain Monte Carlo (AMCMC) of Nott and Kohn (2005) and the standard metropolized Gibbs (Liu, 1996b, MCMC), see Section 1.2. For the MCMC, we draw the number of bits to be flipped from a truncated geometric distribution with mean $k^* = 2$, see Section 1.3.1. However, we could not observe a significant effect of changes in the block updating schemes on the quality of the Monte Carlo estimate. For the AMCMC, we use $\delta = 0.01$ and $\lambda = 0.01$, following the recommendations of Nott and Kohn (2005). We update the estimates $\psi$ and $\mathbf{W}$ every $2 \times 10^5$ iterations of chain. Before we start adapting, we generate $2.5 \times 10^5$ iterations with a metropolized Gibbs kernel (after a discarded burn-in of $2.5 \times 10^4$ iterations).

We run each algorithm 200 times and each time we obtain a Monte Carlo estimate of the marginal probabilities of inclusion of all predictors. We visualize the variation of the estimator by box-plots that show how much the Monte Carlo estimates have varied throughout the 200 runs (Figures 4.4 to 4.9). Here, the white boxes contain 80% of the Monte Carlo results, while the black boxes show the extent of the 20% outliers. For better readability, we add a colored bar up to the smallest estimate we obtained in the test runs; otherwise components with a small variation are hard to see.

The vertical line in the white box indicates the median of the Monte Carlo estimates. The median of the SMC runs correspond very precisely to the results we obtained by running a MCMC algorithm for a few days. Unquestionably, the SMC algorithm is extremely robust; for 200 test runs and for both data sets, the algorithm did not produce a single major outlier in any of the components. This not true for either of the MCMC algorithms. The size of white boxes indicate that adaptive MCMC works quite better than the standard MCMC procedure. However, even the adaptive MCMC method is rather vulnerable to generating outliers. The large black boxes indicate that, for some starting points of the chain, the estimates of some marginal probabilities might be completely wrong.

The outliers, that is the black boxes, in the AMCMC and the MCMC plots are strikingly similar. The adaptive and the standard Markov chains apparently both fall into the same trap, which in turn confirms the intuition that adaption makes a method faster but not more robust against outliers. An adaptive local method is still a local method and does not yield reliable estimates for difficult binary sampling problems. Figure 4.9 suggests that in constrained spaces adaption is difficult and might even have contra-productive effects.

In Tables 4.4 to 4.9, we gather some key performance indicators, each averaged over the 200 runs of the respective algorithms. Note that the time needed to perform $2.5 \times 10^6$

evaluations of $\pi$ is a little less than the running time of the standard MCMC. Thus, even in terms of computational time, the adaptive MCMC can hardly compete with our SMC method, even if evaluations of $\pi$ were at no cost. Note that the time measurements refer to the running time of a pure Python implementation which has been improved significantly since these results were published; see the Appendix on the software for more details.

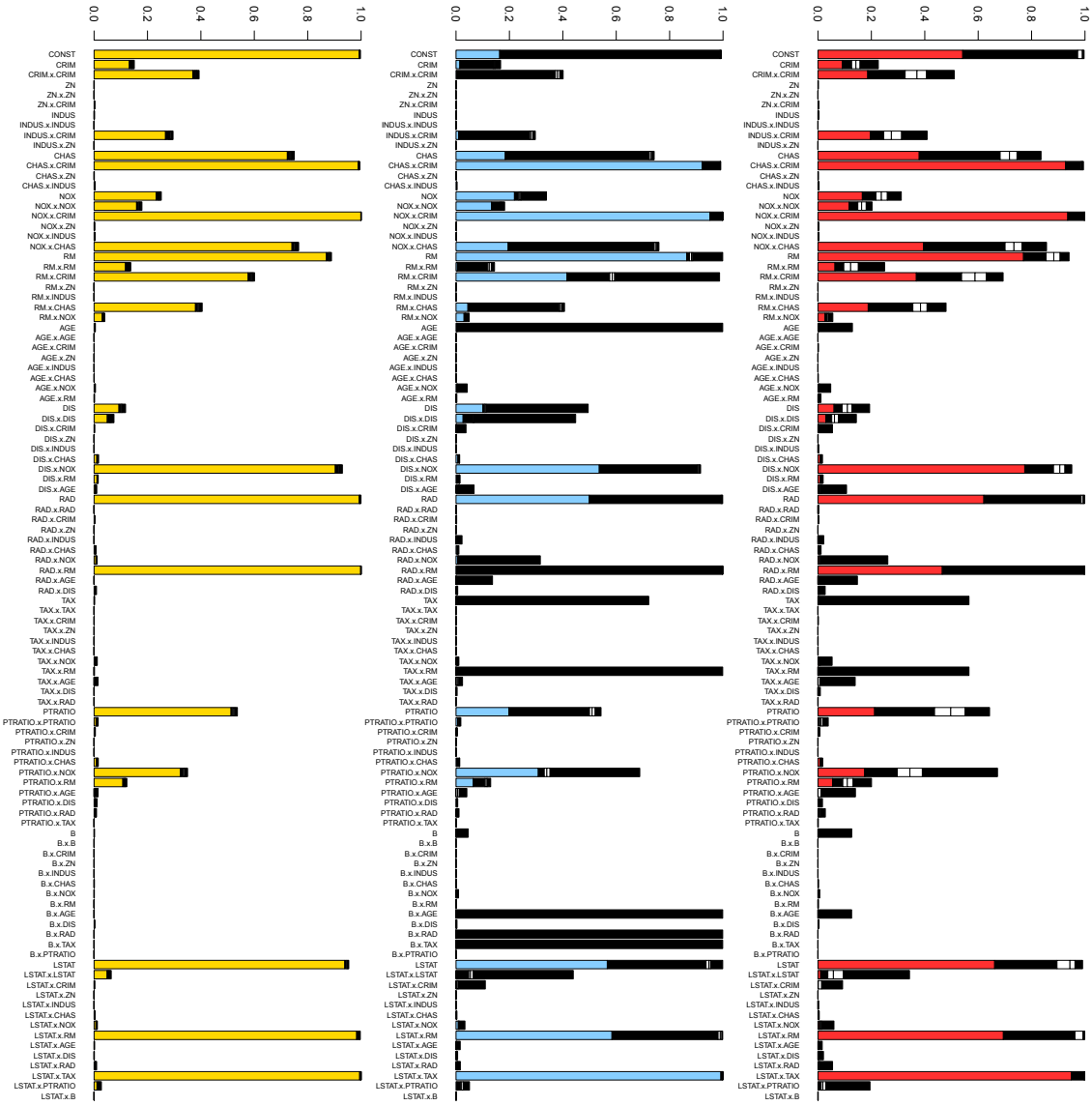
### 4.5.3. Assets and drawbacks

The SMC and the MCMC algorithms both have extensions and numerical speed-ups which make it hard to settle on a fair comparison. Advocates of MCMC methods might criticize that the number of target evaluations is a criterion biased towards the SMC approach, for there are updating schemes which allow for faster computation of the Cholesky decomposition given the decomposition of a neighboring model, see Dongarra et al. (1979, chaps. 8,10). Thus, Markov chains which propose to change one component in each step can evaluate $\pi$ with less effort and perform more evaluations of $\pi$ in the same computational time.

On the other hand, however, the SMC algorithm can be parallelized in the sense that we can, on suitable hardware, run many evaluations of $\pi$ in parallel during the move step, see Procedure 6. No analogue speed-up can be performed in the context of MCMC. Further, SMC methods are more suitable than MCMC to approximate the evidence, that is the normalization constant of the posterior distribution. We can exploit this property to compare, for instance, generalized regression models with different link functions.

Although the numerical results are encouraging, we do not get something for nothing using the SMC sampler. Firstly, the implementation of our algorithm including the logistic conditionals family introduced in Section 3.3 is quite involved compared to standard MCMC algorithms. Secondly, simple MCMC methods are faster than our algorithm while producing results of the same accuracy if the components of the target distribution are nearly independent. Finally, the SMC sampler cannot be used to average out further nuisance parameters but requires a setup where the posterior distribution of the models are available in closed form. In the following Chapter 5, we discuss extensions to the SMC sampler to deal with the latter problem.

**Figure 4.4.:** Boston Housing data set. For details see Section 4.5.1.

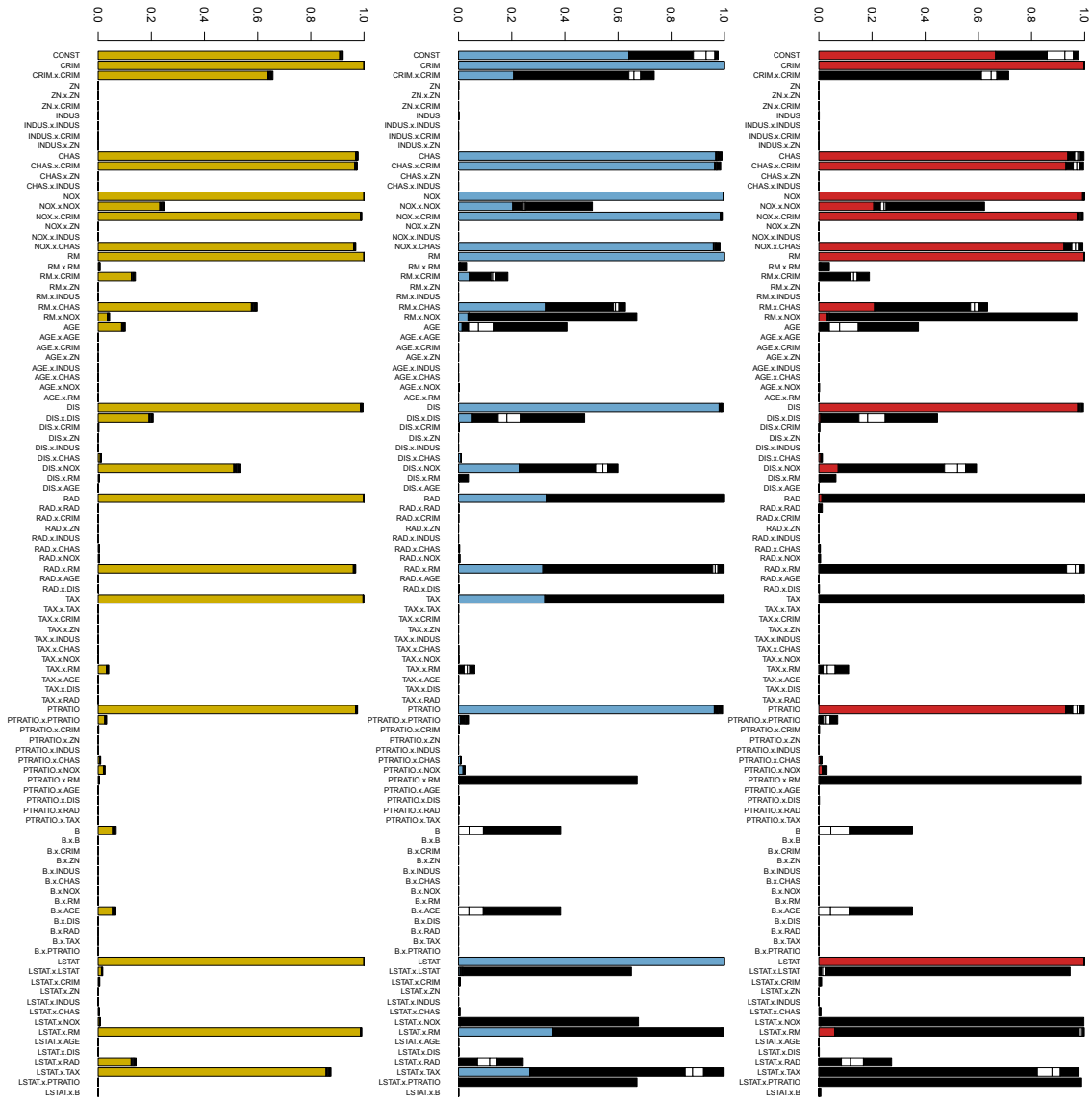


(a) SMC $\sim 1.4 \times 10^6$ evaluations of $\pi$ (b) AMCMC $2.5 \times 10^6$ evaluations of $\pi$ (c) MCMC $2.5 \times 10^6$ evaluations of $\pi$

**Table.** Boston Housing data set. Averaged key indicators complementary to Figure 4.4.

| | SMC | AMCMC | MCMC |
|---|---|---|---|
| computational time | 0 : 36 : 59 h | 4 : 50 : 52 h | 0 : 38 : 06 h |
| evaluations of $\pi$ | $1.36 \times 10^6$ | $2.50 \times 10^6$ | $2.50 \times 10^6$ |
| average acceptance rate | 36.4% | 29.1% | 0.81% |
| length $t$ of the chain $\boldsymbol{x}_t$ | | $7.52 \times 10^7$ | $2.50 \times 10^6$ |
| moves $\boldsymbol{x}_t \neq \boldsymbol{x}_{t-1}$ | | $7.28 \times 10^5$ | $2.07 \times 10^4$ |

**Figure 4.5.:** Boston Housing data set with main effect restrictions. For details see Section 4.5.1.

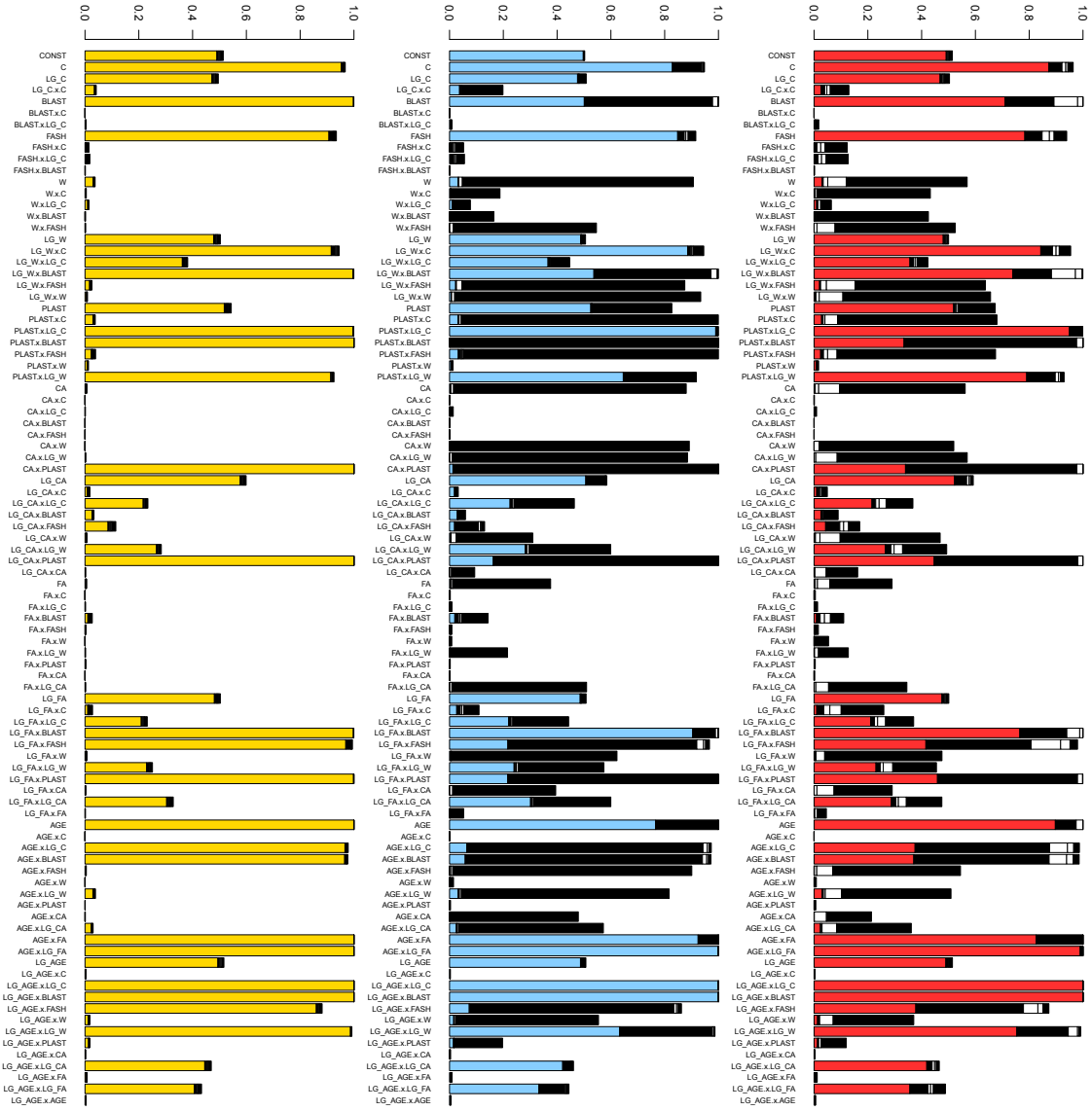


(a) SMC $\sim 1.2 \times 10^6$ evaluations of $\pi$ (b) AMCMC $2.5 \times 10^6$ evaluations of $\pi$ (c) MCMC $2.5 \times 10^6$ evaluations of $\pi$

**Table.** Boston Housing data set with main effect restrictions. Averaged key indicators complementary to Figure 4.5.

| | SMC | AMCMC | MCMC |
|---|---|---|---|
| computational time | 0 : 18 : 05 h | 4 : 33 : 20 h | 0 : 14 : 13 h |
| evaluations of $\pi$ | $1.15 \times 10^6$ | $2.50 \times 10^6$ | $2.50 \times 10^6$ |
| average acceptance rate | 20.79% | 45.4% | 1.20% |
| length $t$ of the chain $\boldsymbol{x}_t$ | | $8.01 \times 10^7$ | $2.50 \times 10^6$ |
| moves $\boldsymbol{x}_t \neq \boldsymbol{x}_{t-1}$ | | $1.13 \times 10^6$ | $2.96 \times 10^4$ |

**Figure 4.6.:** Concrete Compressive Strength data set. For details see Section 4.5.1.

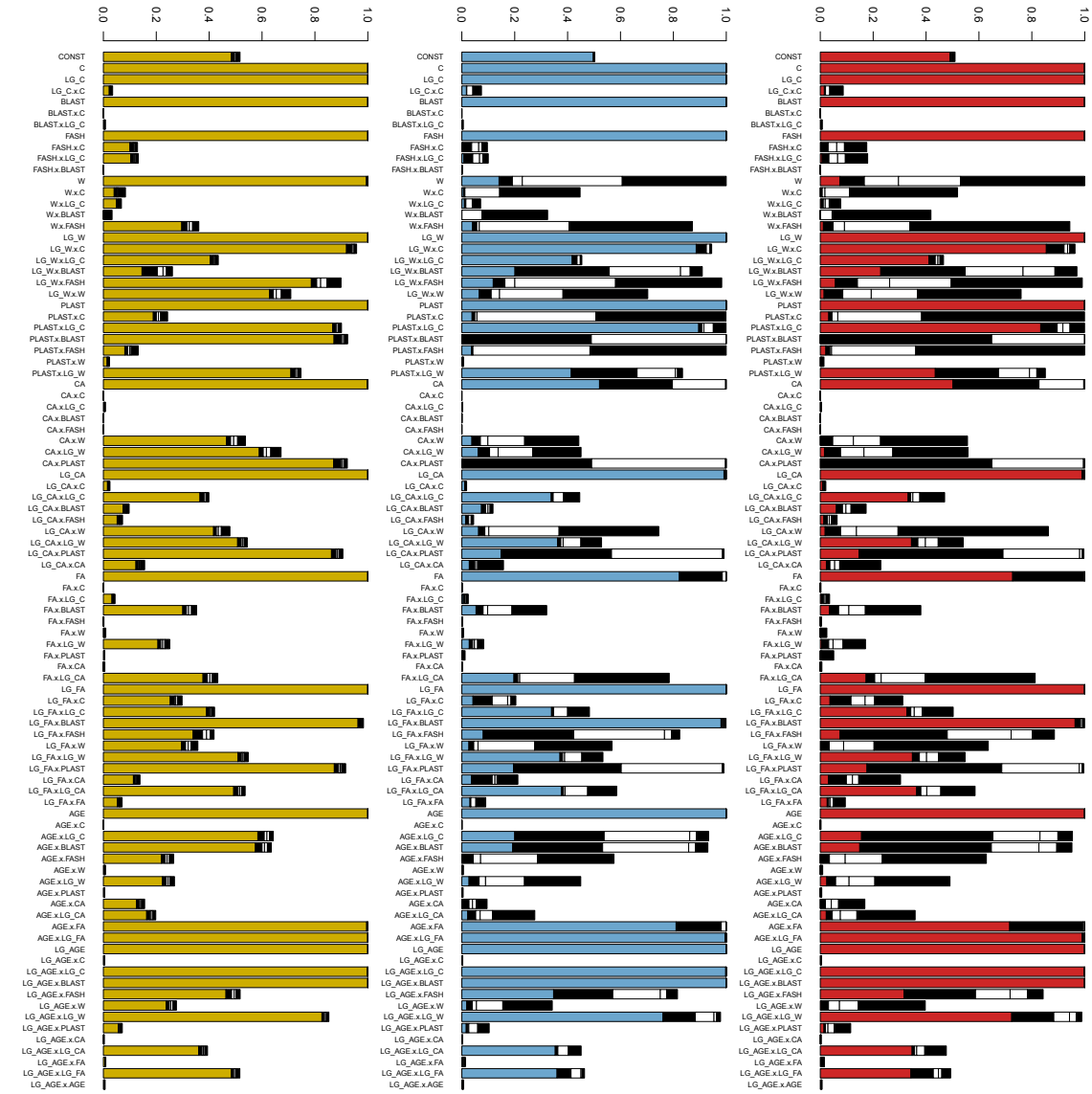


(a) SMC $\sim 1.2 \times 10^6$ evaluations of $\pi$ (b) AMCMC $2.5 \times 10^6$ evaluations of $\pi$ (c) MCMC $2.5 \times 10^6$ evaluations of $\pi$

**Table.** Concrete Compressive Strength data set. Averaged key indicators complementary to Figure 4.6.

| | SMC | AMCMC | MCMC |
|---|---|---|---|
| computational time | 0 : 29 : 01 min | 2 : 02 : 06 min | 0 : 43 : 17 min |
| evaluations of $\pi$ | $1.19 \times 10^6$ | $2.50 \times 10^6$ | $2.50 \times 10^6$ |
| average acceptance rate | 30.7% | 70.4% | 7.20% |
| length $t$ of the chain $\boldsymbol{x}_t$ | | $2.43 \times 10^7$ | $2.50 \times 10^6$ |
| moves $\boldsymbol{x}_t \neq \boldsymbol{x}_{t-1}$ | | $1.76 \times 10^6$ | $1.79 \times 10^5$ |

**Figure 4.7.:** Concrete Compressive Strength data set with main effect restrictions. For details see Section 4.5.1.

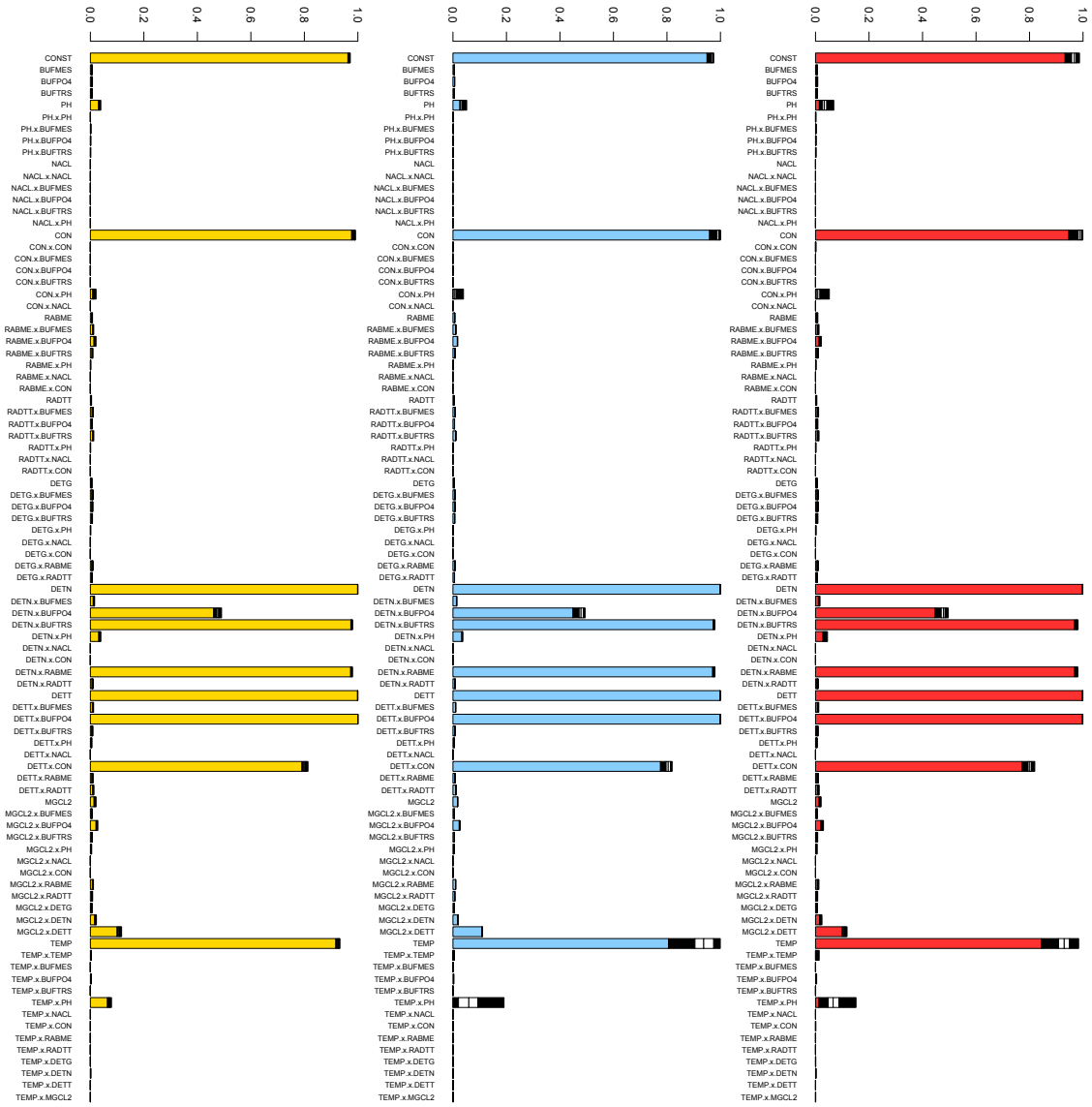


(a) SMC $\sim 2.4 \times 10^6$ evaluations of $\pi$ (b) AMCMC $2.5 \times 10^6$ evaluations of $\pi$ (c) MCMC $2.5 \times 10^6$ evaluations of $\pi$

**Table.** Concrete Compressive Strength data set with main effect restrictions. Averaged key indicators complementary to Figure 4.7.

| | SMC | AMCMC | MCMC |
|---|---|---|---|
| computational time | 0 : 43 : 01 min | 2 : 29 : 16 min | 0 : 41 : 48 min |
| evaluations of $\pi$ | $2.42 \times 10^6$ | $2.50 \times 10^6$ | $2.50 \times 10^6$ |
| average acceptance rate | 30.98% | 61.1% | 5.31% |
| length $t$ of the chain $\boldsymbol{x}_t$ | | $2.72 \times 10^7$ | $2.50 \times 10^6$ |
| moves $\boldsymbol{x}_t \neq \boldsymbol{x}_{t-1}$ | | $1.53 \times 10^6$ | $1.32 \times 10^5$ |

**Figure 4.8.:** Protein data set. For details see Section 4.5.1.

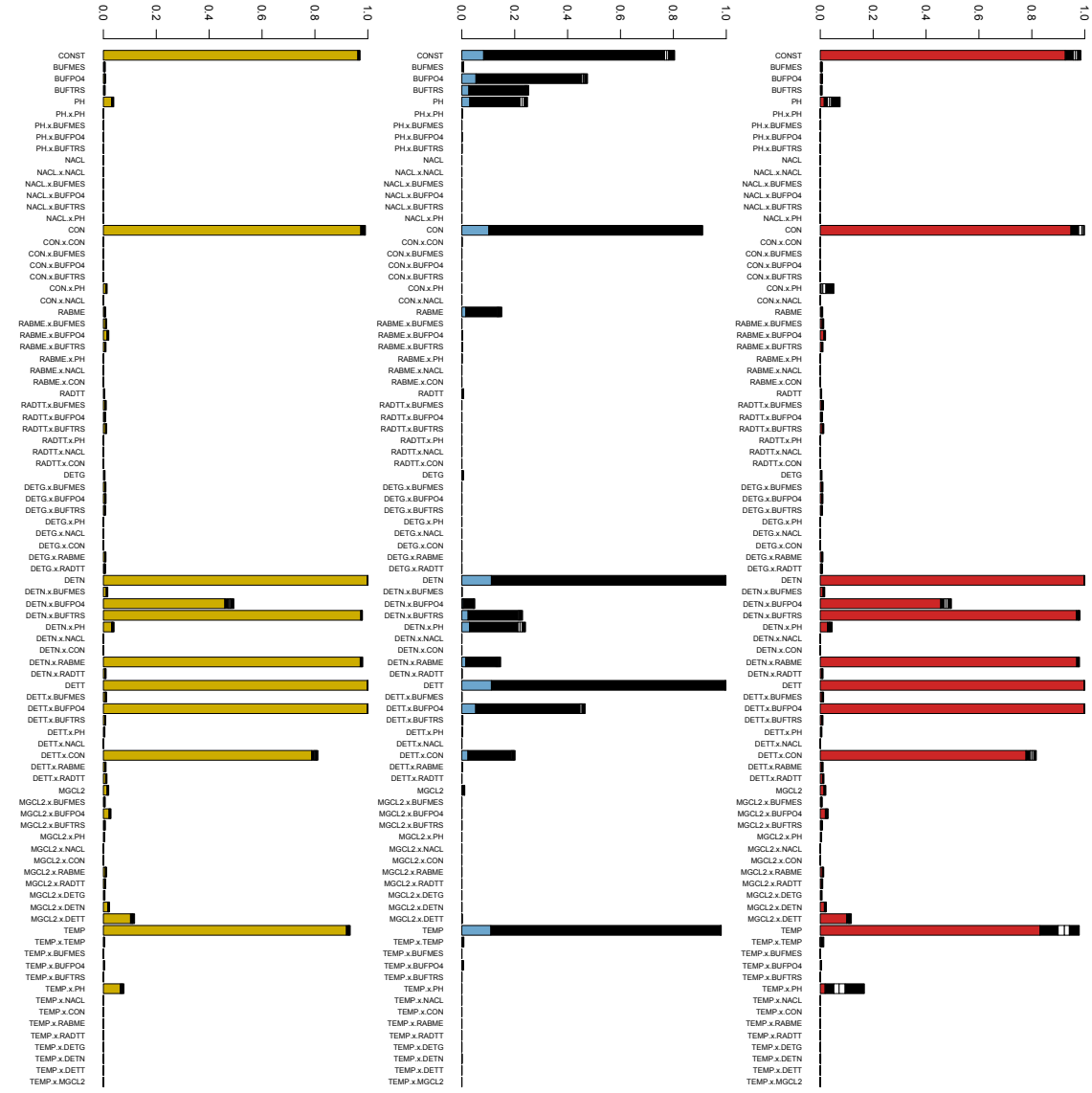



(a) SMC $\sim 6.1 \times 10^5$ evaluations of $\pi$ (b) AMCMC $2.5 \times 10^6$ evaluations of $\pi$ (c) MCMC $2.5 \times 10^6$ evaluations of $\pi$

**Table.** Protein data set. Averaged key indicators complementary to Figure 4.8.

| | SMC | AMCMC | MCMC |
|---|---|---|---|
| computational time | 0 : 14 : 55 min | 3 : 58 : 32 min | 0 : 29 : 38 min |
| evaluations of $\pi$ | $6.17 \times 10^5$ | $2.50 \times 10^6$ | $2.50 \times 10^6$ |
| average acceptance rate | 30.7% | 60.7% | 1.20% |
| length $t$ of the chain $\boldsymbol{x}_t$ | | $9.19 \times 10^7$ | $2.50 \times 10^6$ |
| moves $\boldsymbol{x}_t \neq \boldsymbol{x}_{t-1}$ | | $1.51 \times 10^6$ | $3.03 \times 10^5$ |

**Figure 4.9.:** Protein data set with main effect restrictions. For details see Section 4.5.1.



(a) SMC $\sim 6.1 \times 10^5$ evaluations of $\pi$ (b) AMCMC $2.5 \times 10^6$ evaluations of $\pi$ (c) MCMC $2.5 \times 10^6$ evaluations of $\pi$

**Table.** Protein data set with main effect restrictions. Averaged key indicators complementary to Figure 4.9.

| | SMC | AMCMC | MCMC |
|---|---|---|---|
| computational time | 0 : 14 : 45 min | 3 : 32 : 06 min | 0 : 30 : 21 min |
| evaluations of $\pi$ | $6.19 \times 10^5$ | $2.50 \times 10^6$ | $2.50 \times 10^6$ |
| average acceptance rate | 26.65% | 22.3% | 1.20% |
| length $t$ of the chain $\boldsymbol{x}_t$ | | $1.07 \times 10^8$ | $2.50 \times 10^6$ |
| moves $\boldsymbol{x}_t \neq \boldsymbol{x}_{t-1}$ | | $5.56 \times 10^6$ | $3.03 \times 10^5$ |

# 5. Bayesian variable selection for binary response models


## Resumé

*Ce chapitre propose des idées pour étendre les méthodes de Monte Carlo séquentielles à la sélection bayésienne de variables dans le contexte des modèles linéaires généralisés à réponse binaire comme les modèles de régression logistique ou probit. Dans ce cas, la distribution a posteriori n'est pas disponible sous forme fermée, et les paramètres du modèle doivent être marginalisés à l'aide soit d'approximations, soit d'approches pseudo-marginales afin d'appliquer l'algorithme de Monte Carlo séquentiel. Par analogie au chapitre 4, plusieurs instances de test sur données réelles sont construites et l'échantillonneur de Monte Carlo séquentiel est comparé à l'échantillonneur automatique générique (Green, 2003) qui est une méthode de Monte Carlo à chaîne de Markov transdimensionnel.*


## 5.1. Introduction

We discuss the sequential Monte Carlo (SMC) sampler developed in Chapter 2 can be extended to Bayesian variable selection in the context of generalized linear models with binary response. Compared to variable selection in normal linear models treated in the preceding chapter, we face the problem that the marginal likelihood is not available in closed-form.

Let $Y$ denote the random quantity of interest or *response* and $\boldsymbol{Z}$ a $d$-dimensional vector of *covariates* or *predictors*. A generalized linear model assumes that $Y$ conditional on $\boldsymbol{Z} = \boldsymbol{z}$ has a density or mass function from the exponential family which can be written in terms of a linear predictor and a link function $\mu$ such that

$$\mathbb{E}\left(Y \mid \boldsymbol{Z} = \boldsymbol{z}\right) = \mu(\beta_0 + \boldsymbol{\beta}^\intercal \boldsymbol{z}),$$

see McCullagh and Nelder (1989) for details. For binary response variables, the typical model is

$$h(y, \boldsymbol{z}) = \mu(\beta_0 + \boldsymbol{\beta}^\intercal \boldsymbol{z})^y [1 - \mu(\beta_0 + \boldsymbol{\beta}^\intercal \boldsymbol{z})]^{1-y} \tag{5.1}$$

where $\mu$ is sigmoid, log-concave and twice differentiable; important special cases are the *probit* regression for $\mu(x) = (2\pi)^{-1/2} \int_{-\infty}^{x} \exp(-y^2/2) dy$ and the *logistic* regression for $\mu(x) = 1/[1 + \exp(-x)]$.

We denote by $n$ the number of observation, by $\boldsymbol{y} \in \mathbb{R}^n$ the vector of observed explained variables and by $\mathbf{Z} \in \mathbb{R}^{n \times d}$ the design matrix of observed explanatory variables. We identify each regression model with a binary vector $\boldsymbol{\gamma} \in \mathbb{B}^d$ where the predictor $Z_i$ is in the model if and only if $\gamma_i = 1$. For convenience of notation, we write $\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} = (\beta_0, \boldsymbol{\beta}_{\boldsymbol{\gamma}}^\intercal)^\intercal$ for the vector of all regression parameters of the model indicated by $\boldsymbol{\gamma}$.

### 5.1.1. Selection criteria

The remarks on penalized likelihood criteria made in Section 4.1.3 also apply in the context of generalized linear models. However, unlike for linear normal models there is no closed-form expression for the maximum likelihood estimators and maximization has to be done numerically as described in Section 5.2.1. The convex optimization techniques mentioned in Section 4.1.4 may also be extended to generalized linear models with convex penalties which includes (5.1). For details, we refer to Friedman et al. (2010) and citations therein.

### 5.1.2. Bayesian variable selection

In the following, we only consider Bayesian approaches to variable selection where the selection criterion is the posterior distribution on the model space. The discussion on the choice of prior distributions on the model space in Section 4.3 equally applies to generalized linear models. The additional difficulty with respect to variable selection in the context of normal linear models is the lack of conjugate priors which would allow to obtain the marginal likelihood in closed-form. We denote the likelihood by

$$\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) := \textstyle\prod_{k=1}^{n} h(y_k, \boldsymbol{z}_k \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}),$$

and for suitable prior distributions on the regression parameters and the model space, we obtain an unnormalized posterior distribution via Bayes' Theorem

$$\pi(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \mid \boldsymbol{y}, \mathbf{Z}) \propto \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) p(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma}) p(\boldsymbol{\gamma}).$$

We discuss the choice of the prior on regression parameters in Section 5.2.2 but limit the analysis to priors which yield posterior distributions that are log-concave in $\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ given $\boldsymbol{\gamma}$. The computational challenge is to provide an estimate of

$$\pi(f \mid \boldsymbol{y}, \mathbf{Z}) = \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} f(\boldsymbol{\gamma}) \int_{\mathbb{R}^{|\boldsymbol{\gamma}|+1}} \pi(\boldsymbol{\gamma}, \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} \mid \boldsymbol{y}, \mathbf{Z}) d\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \tag{5.2}$$

where $f$ might be any quantity of interest. There are solution based on transdimensional Markov chain Monte Carlo (MCMC) sampling schemes which allow to sample from the joint distribution of the model and the regression parameters. We briefly review this approach in Section 5.3. In order to make the SMC sampler work for this kind of problem, we may compute or approximate the marginal likelihood

$$\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma}) = \int_{\mathbb{R}^{|\boldsymbol{\gamma}|+1}} \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) p(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma}) d\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$$

every time we evaluate the posterior distribution $\pi(\cdot \mid \boldsymbol{y}, \mathbf{Z})$ on the model space and proceed as in the preceding chapter on normal linear models.

## 5.2. Marginal likelihood

In the context of linear normal regression models, we can calculate a closed-form expression of the marginal likelihood up to a constant for a judicious choice of the prior distributions, see Section 4.2.1. This is not possible for generalized linear models with binary response. In order to compute the marginal likelihood

$$\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma}) = \int \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) p(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}) d\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$$

we might either resort to some approximation scheme or use a Monte Carlo estimate.

### 5.2.1. Maximum likelihood

We briefly review how to compute the mode of the likelihood function which is an important ingredient of both the approximation and the Monte Carlo scheme. For simplicity, we assume that $\mu$ is log-concave with an odd second derivative $\mu''$ which ensure that the likelihood function is concave. In other words, let $\mu \colon \mathbb{R} \to [0, 1]$ be a twice differentiable increasing bijection which satisfies

$$-\frac{[\mu'(x)]^2}{1 - \mu(x)} \leq \mu''(x) \leq \frac{[\mu'(x)]^2}{\mu(x)}, \quad x \in \mathbb{R}. \tag{5.3}$$

A sufficient condition for (5.3) is that $\mu'$ is an even log-concave density function which implies that $\mu$ is also log-concave and the second derivative $\mu''$ is odd. Popular examples are the *logistic* and *probit* link functions.

We let $\boldsymbol{y}$ denote the vector of observations, $\mathbf{Z}_{\boldsymbol{\gamma}}$ the design matrix and let $\boldsymbol{\gamma}$ be the binary vector encoding the model. For ease of notation, we define

$$\eta_k := \beta_0 + \boldsymbol{z}_{k,\boldsymbol{\gamma}}\boldsymbol{\beta}_{\boldsymbol{\gamma}}$$

for the linear predictor of the $k$th observation and let $\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} = (\beta_0, \boldsymbol{\beta}_{\boldsymbol{\gamma}}^{\intercal})^{\intercal}$ denote the vector of the regression parameters including the intercept. The log-likelihood function of the generalized linear model is

$$\log \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) = \sum_{k=1}^{n} \left( y_k \log[\mu(\eta_k)] + (1 - y_k) \log[1 - \mu(\eta_k)] \right),$$

the gradient is

$$s_{\boldsymbol{\gamma}}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}) := \frac{\partial \log \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma})}{\partial \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}} = \sum_{k=1}^{n} (1, \boldsymbol{z}_{k,\boldsymbol{\gamma}}) \left[ y_k \frac{\mu'(\eta_k)}{\mu(\eta_k)} - (1 - y_k) \frac{\mu'(\eta_k)}{1 - \mu(\eta_k)} \right],$$

and the Hessian is

$$\begin{aligned} \frac{\partial^2 \log \log \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma})}{\partial \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} \partial \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{\intercal}} = \sum_{k=1}^{n} (1, \boldsymbol{z}_{k,\boldsymbol{\gamma}})^{\intercal} (1, \boldsymbol{z}_{k,\boldsymbol{\gamma}}) \Bigg[ & y_k \left[ \frac{\mu''(\eta_k)}{\mu(\eta_k)} - \frac{[\mu'(\eta_k)]^2}{[\mu(\eta_k)]^2} \right] \\ & + (1 - y_k) \left[ -\frac{\mu''(\eta_k)}{1 - \mu(\eta_k)} - \frac{[\mu'(\eta_k)]^2}{[1 - \mu(\eta_k)]^2} \right] \Bigg]. \end{aligned}$$

The first order condition $s_{\boldsymbol{\gamma}}^{\epsilon}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}) = \mathbf{0}$ is typically solved via Newton Raphson iterations

$$\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{(t+1)} = \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{(t)} + F_{\boldsymbol{\gamma}}^{-1}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{(t)}) s_{\boldsymbol{\gamma}}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{(t)})$$

for some suitable starting point $\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{(0)} \in \mathbb{R}^p$ where

$$F_{\boldsymbol{\gamma}}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}) := -\frac{\partial^2 \log \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma})}{\partial \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} \partial \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{\intercal}} \geq 0, \quad \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} \in \mathbb{R}^{|\boldsymbol{\gamma}|+1}$$

denotes the observed Fisher information matrix. Note that condition (5.3) ensures that $F_{\boldsymbol{\gamma}}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}})$ is positive semi-definite and the likelihood function therefore log-concave. This guarantees the uniqueness but not the existence of the maximizer since the data might suffer from complete or quasi-complete separation (Albert and Anderson, 1984) which would cause the likelihood function to be monotonic. However, we can assure that the likelihood function is strictly log-concave by assigning a suitable prior distribution to the regression parameter $\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$.

### 5.2.2. Prior on the regression parameters

Firth (1993) recommends the Jeffreys prior for its bias reduction which can conveniently be implemented via a data adjustment scheme (Kosmidis and Firth, 2009). For the sake of simplicity, we work with a simple multivariate normal prior $p = \mathcal{N}(\mathbf{0}, \tau\Sigma_{\boldsymbol{\gamma}})$ for a dispersion parameter $\tau > 0$ such that, up to a constant, the log-posterior distribution is the log-likelihood function plus a quadratic penalty term which gives

$$\pi(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \mid \boldsymbol{y}, \mathbf{Z}) \propto \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}) \exp\left[-\frac{1}{2\tau}\tilde{\boldsymbol{\beta}}^{\intercal}\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}^{-1}\tilde{\boldsymbol{\beta}}\right].$$

The score and Fisher matrix under the prior are

$$s_{\boldsymbol{\gamma}}^{p}(\tilde{\boldsymbol{\beta}}) := s_{\boldsymbol{\gamma}}(\tilde{\boldsymbol{\beta}}) - \tau^{-1}\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}^{-1}\tilde{\boldsymbol{\beta}}, \quad F_{\boldsymbol{\gamma}}^{p}(\tilde{\boldsymbol{\beta}}) := F_{\boldsymbol{\gamma}}(\tilde{\boldsymbol{\beta}}) + \tau^{-1}\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}^{-1}.$$

We should choose the dispersion parameter $\tau$ small enough to ensure numerical stability of the maximization procedure but large enough to avoid an unnecessary shrinkage effect. The normal prior ensures that likelihood function remains concave and maximization is fairly straightforward; using heavy-tailed priors like student's t distribution we would loose this property.

### 5.2.3. Laplace approximation

Let $\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \cdot, \boldsymbol{\gamma})$ denote the likelihood with respect to the regression coefficients and let

$$\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{*} := \operatorname{argmax}_{\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} \in \mathbb{R}^{|\boldsymbol{\gamma}|+1}} \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) p(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}})$$

be the penalized maximum-likelihood estimator under the multivariate normal prior $p$. A second order Taylor expansion of $\log[\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \cdot, \boldsymbol{\gamma})p]$ around $\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{*}$ yields the approximation

$$\log[\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) p(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}})] \approx \log \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{*}, \boldsymbol{\gamma}) - \frac{1}{2}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} - \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{*})^{\intercal} F_{\boldsymbol{\gamma}}^{p}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{*})(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} - \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{*})$$

which allows to approximate the marginal likelihood by

$$\hat{\mathcal{L}}_{\text{L}}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma}) := \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{*}, \boldsymbol{\gamma})(2\pi)^{(|\boldsymbol{\gamma}|+1)/2} \det[\mathbf{F}_{\boldsymbol{\gamma}}^{p}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^{*})]^{-1/2},$$

where the Fisher matrix under the prior $F_{\boldsymbol{\gamma}}^{p}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}})$ is defined in the preceding section.

### 5.2.4. Pseudo-marginal sampler

The SMC sampler is only designed to sample from distribution with support $\mathbb{B}^d$, but we might compute an unbiased Monte Carlo estimate of the marginal distribution each

time we evaluate the posterior distribution. Since the regression parameters are a priori assumed to be normal distributed, we can design an importance sampling (IS) estimator using the student's t approximation

$$\varphi_\nu^t(\boldsymbol{x}) = \frac{\Gamma[(\nu+|\boldsymbol{\gamma}|+1)/2]}{\Gamma[\nu/2](\nu\pi)^{(|\boldsymbol{\gamma}|+1)/2}\left|\mathbf{F}_{\boldsymbol{\gamma}}^p(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^*)\right|^{1/2}\left[1+\frac{1}{\nu}(\boldsymbol{x}-\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^*)^\intercal\mathbf{F}_{\boldsymbol{\gamma}}^p(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^*)(\boldsymbol{x}-\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^*)\right]^{(\nu+|\boldsymbol{\gamma}|+1)/2}}$$

where $\nu \in \mathbb{N}$ denotes the degrees of freedom, $\boldsymbol{\beta}_{\boldsymbol{\gamma}}^*$ the maximum likelihood estimator and $\mathbf{F}_{\boldsymbol{\gamma}}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^*)$ the observed Fisher information under the prior $p$; see section 5.2.3. For a sample $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ from the instrumental distribution $\mathcal{T}_\nu(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^*, \mathbf{F}_{\boldsymbol{\gamma}}^{-1}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^*))$ we obtain the IS estimator

$$\hat{\mathcal{L}}_{\text{IS}}^m(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma}) = \frac{1}{m}\sum_{k=1}^m \frac{\mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{v}_k, \boldsymbol{\gamma})p(\boldsymbol{v}_k)}{\varphi_\nu^t[\boldsymbol{v}_k \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^*, \mathbf{F}_{\boldsymbol{\gamma}}^{-1}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}^*)]}, \tag{5.4}$$

which converges $\hat{\mathcal{L}}_{\text{IS}}^m(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma}) \overset{m\to\infty}{\longrightarrow} \mathcal{L}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma})$ a.s. by virtue of the law of large numbers, see Section 1.1.2.

Andrieu and Roberts (2009) generalize the GIMH algorithm by Beaumont (2003) and show that the MCMC estimator remains valid even if the density of the target function in the acceptance probability of the Metropolis-Hastings kernel (1.7) is replaced by an unbiased estimator. Chopin et al. (2011) propose an SMC sampling scheme based on the same rationale,

$$\hat{\pi}_{\text{SMC}}^{n,m}(f) = \sum_{k=1}^n w_{k,\tau}^m f(\boldsymbol{X}_{k,\tau}),$$

where $\hat{\pi}_{\text{SMC}}^{n,m}(f) \overset{n,m\to\infty}{\longrightarrow} \pi(f)$ a.s. which justifies the pseudo-marginal approach in the context of the sampler proposed in Chapter 2.

The practical question arises, how many samples one should use for the IS estimators and how many particles for the SMC sampler. It seems difficult to provide general guidance. The number of samples necessary for the IS estimator $\hat{\mathcal{L}}_{\text{IS}}^m(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma})$ to provide a certain precision depends on the model $\boldsymbol{\gamma}$, and we propose to choose $m$ such that the effective sample size (ESS) $\eta_{\text{IS}}^m$ of the IS estimator reaches at least some target value $\eta_{\text{IS}}^*$ at the final stage of the SMC sampler.

If $(\varrho_t)_{t\in\mathbb{N}}$ denotes the annealing schedule defined in Section 2.2.2, we choose the sample size of the IS estimator at time $t$ such that the ESS is at least $\varrho_t\eta_{\text{IS}}^*$. In other words, the target ESS of the IS estimator increases during the run of the algorithm. The rationale behind this choice is that less precision is necessary in the early stage of the annealing SMC. Numerical experiments show that using the full precision $\eta_{\text{IS}}^*$ for the whole run of the SMC sampler considerably slows down the algorithm but hardly improves the estimator.

### 5.2.5. Corrected Laplace sampler

Computing an IS estimator $\hat{\pi}(\boldsymbol{\gamma} \mid \boldsymbol{y}, \mathbf{Z})$ for each evaluation of the posterior distribution is computationally quite costly. A faster alternative is to run the SMC sampler with respect to the Laplace approximation derived in Section 5.2.3 to obtain a sample

$$(\boldsymbol{X}_1, \dots, \boldsymbol{X}_n) \sim \hat{\pi}_{\text{L}}(\boldsymbol{\gamma} \mid \boldsymbol{y}, \mathbf{Z}) \propto \hat{\mathcal{L}}_{\text{L}}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma}) p(\boldsymbol{\gamma}).$$

Using the same ideas as developed in the preceding section on the pseudo-marginal approach, we may compute an IS estimator (5.4) for the marginal likelihood $\hat{\mathcal{L}}^m_{\text{IS}}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{x}_k)$ for all $k \in N$ and finally construct an IS for the posterior distribution

$$\hat{\pi}^{n,m}_{\text{IS}}(f) := \frac{\sum_{k=1}^n f(\boldsymbol{X}_k) w^m_{\text{IS}}(\boldsymbol{X}_k)}{\sum_{k=1}^n w^m_{\text{IS}}(\boldsymbol{X}_k)}, \quad w^m_{\text{IS}}(\boldsymbol{\gamma}) := \frac{\hat{\mathcal{L}}^m_{\text{IS}}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma})}{\hat{\mathcal{L}}_{\text{L}}(\boldsymbol{y}, \mathbf{Z} \mid \boldsymbol{\gamma})}.$$

Naturally, this approach does not depend on the SMC sampler, but the sample from $\hat{\pi}_{\text{L}}(\boldsymbol{\gamma} \mid \boldsymbol{y}, \mathbf{Z})$ may also come from a thinned Markov chain or other sampling schemes.

## 5.3. Transdimensional Markov chain Monte Carlo

### 5.3.1. Reversible jumps

If there is a closed-form expression for the integrated likelihood, the posterior distribution is solely defined on a binary space and standard MCMC tools introduced in Chapter 1 are straightforward to apply. In the case of variable selection for generalized linear models, however, the MCMC procedure has to be defined on the joined space of the model and the regression coefficients.

The typical way to deal with joined distributions $\pi(\boldsymbol{\theta}, \boldsymbol{\gamma})$ defined on $\mathbb{R}^{d+1} \times \mathbb{B}^d$ is Gibbs sampling where one alternates sampling from the full conditional distributions, that is $\pi(\boldsymbol{\theta} \mid \boldsymbol{\gamma})$ and $\pi(\boldsymbol{\gamma} \mid \boldsymbol{\theta})$. In the case of variable selection, however, the model $\boldsymbol{\gamma}$ is completely defined by the vector of regression parameters $\boldsymbol{\theta} = \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$, and the expression $\pi(\boldsymbol{\gamma} \mid \tilde{\boldsymbol{\beta}}_{\boldsymbol{\gamma}})$ is therefore not meaningful. The appropriate state space for the variable selection problem is $\cup_{\boldsymbol{\gamma} \in \mathbb{B}^d} (\mathbb{R}^{|\boldsymbol{\gamma}|+1} \times \{\boldsymbol{\gamma}\})$ and MCMC methods dealing with these non-standard spaces are referred to as *transdimensional* Markov chain Monte Carlo.

Green (1995) first proposes a solution called *reversible jump* MCMC which introduces a diffeomorphism between models of different dimensions which have to verify a dimension-matching condition to ensure detailed balance. This allows to derive the usual

Metropolis-Hastings acceptance ratio which only involves the Jacobian of the between-models diffeomorphism which comes from the standard change-of-variables formula. For further details, we refer to Green (2003) who provides a constructive representation of this idea in terms of auxiliary random variables.

The major practical problem of reversible jump MCMC is the lack of guidance on how to construct the jump proposals and insufficient tuning is known to result in acceptance probabilities which are prohibitively low. Brooks et al. (2003) elaborate a series of techniques to construct jump functions and saturation schemes. Holmes and Held (2006) propose an extension of the probit data augmentation approach by Albert and Chib (1993) to logistic regression. The advantage of the data augmentation scheme is that $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ can be updated jointly conditional on the auxiliary variables which avoids the problem of transdimensional moves. For a recent comparison of methods see Lamnisos et al. (2009).

### 5.3.2. The automatic generic sampler

We briefly review a reversible jump MCMC scheme proposed by Green (2003) as *automatic generic sampler*. Reversible jump type algorithms are known to need some tuning to provide efficient kernels for a particular problem, and Green (2003) introduces the automatic generic sampler as a generic approach which works particularly well if the regression parameters are close to normality. In Section 5.4, we compare the automatic generic sampler to the SMC sampler combined with the pseudo-marginal technique.

The automatic generic sampler is summarized in Algorithm 11. The auxiliary kernel $q$ on the model space performs a swap move between two uniformly chosen components with probability 1/3; it changes a uniformly chosen component with probability 2/3. As before, we denote by $\tilde{\boldsymbol{\beta}}^*_{\boldsymbol{\gamma}}$ the maximum-likelihood under the prior and let $\mathbf{C}^*_{\boldsymbol{\gamma}}$ be the Cholesky decomposition $\mathbf{C}^*_{\boldsymbol{\gamma}}(\mathbf{C}^*_{\boldsymbol{\gamma}})^\intercal = [\mathbf{F}^p_{\boldsymbol{\gamma}}(\tilde{\boldsymbol{\beta}}^*_{\boldsymbol{\gamma}})]^{-1}$ of the inverse Fisher matrix at the mode. $\mathcal{T}_\nu$ denotes student's t distribution with $\nu \in \mathbb{N}$ degrees of freedom and $\varphi^t_\nu$ its mass function.

## 5.4. Numerical experiments

For our numerical examples, we assume the regression parameters to be a priori independent, that is $\boldsymbol{\Sigma}_{\boldsymbol{\gamma}} = \tau \mathbf{I}_{|\boldsymbol{\gamma}|}$ with dispersion parameter $\tau = n$ where $n$ is the number of observations. We use the prior distribution on the model space described in Section

**Algorithm 11:** Automatic generic sampler

**Input**: $f\colon \mathbb{B}^d \to \mathbb{R}$

$\boldsymbol{x}_0 \leftarrow \boldsymbol{X}_0 \sim p$, $\boldsymbol{u} \leftarrow \boldsymbol{U} \sim \mathcal{T}_\nu(\mathbf{0}, \mathbf{I}_{|\boldsymbol{x}_0|})$, $\tilde{\boldsymbol{\beta}}_{\boldsymbol{x}_0} \leftarrow \hat{\boldsymbol{\beta}}^*_{\boldsymbol{x}_0} + \mathbf{C}^*_{\boldsymbol{x}_0}\boldsymbol{u}$

**for** $k = 0$ **to** $n$ **do**

  $\boldsymbol{x}' \sim q(\cdot \mid \boldsymbol{x}_k)$, $\boldsymbol{u}' \leftarrow \boldsymbol{u}$

  **if** $|\boldsymbol{x}_k| > |\boldsymbol{x}'|$ **then** $v \leftarrow u_{|\boldsymbol{x}'|}$, $\boldsymbol{u}' \leftarrow \boldsymbol{u}_{1:|\boldsymbol{x}'|-1}$

  **if** $|\boldsymbol{x}_k| < |\boldsymbol{x}'|$ **then** $v \leftarrow V \sim \mathcal{T}_\nu(0,1)$, $\boldsymbol{u}' \leftarrow (\boldsymbol{u}^\intercal, v)^\intercal$

  $\tilde{\boldsymbol{\beta}}' \leftarrow \tilde{\boldsymbol{\beta}}^*_{\boldsymbol{x}'} + \mathbf{C}^*_{\boldsymbol{x}'}\mathbf{P}\boldsymbol{u}'$

  $$\alpha \leftarrow \frac{\pi(\boldsymbol{x}', \tilde{\boldsymbol{\beta}}')}{\pi(\boldsymbol{x}_k, \tilde{\boldsymbol{\beta}}_{\boldsymbol{x}_k})} \frac{|\mathbf{C}^*_{\boldsymbol{x}'}|}{|\mathbf{C}^*_{\boldsymbol{x}_k}|} \cdot \begin{cases} [\varphi^t_\nu(v)]^{-1} & \text{if } |\boldsymbol{x}_k| > |\boldsymbol{x}'| \\ 1 & \text{if } |\boldsymbol{x}_k| = |\boldsymbol{x}'| \\ \varphi^t_\nu(v) & \text{if } |\boldsymbol{x}_k| < |\boldsymbol{x}'| \end{cases}$$

  **if** $\alpha > U \sim \mathcal{U}_{[0,1]}$ **then**

    $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}'$, $\boldsymbol{u} \leftarrow \mathbf{P}\boldsymbol{u}'$

  **else**

    $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k$

  **end**

**end**

**return** $(n+1)^{-1} \sum_{k=0}^{n} f(\boldsymbol{x}_k)$

4.3.1 where the a priori expected model size $\overline{d} \in D$ was fixed to some reasonable value and the maximum model size was chosen $d^* = 2\overline{d}$.

### 5.4.1. Construction of test instances

For testing, we created variable selection problems with binary response from datasets which are freely available at the UCI Machine Learning Repository. In the following, we briefly describe the variable selection problems composed for our numerical experiments.

**Australian Credit Approval** The first example comes from a credit card application, originally treated by Quinlan (1987), where the goal is to determine the credit worthiness from a set of predictors. The attribute names and values have been altered to protect the confidentiality of the data. Missing values had been replaced by the modes of the corresponding attributes. The original data set has 690 observations and 14 predictors where we introduced additional dummy variables for the categorical factors $V4, V5, V6$ and $V12$ which yields a total of 34 covariates.

**Wisconsin Prognostic Breast Cancer** The second example is concerned with the problem of predicting whether a breast cancer is recurrent or not recurrent before 24 months. In a series of publications, Wolberg et al. (1995) analyzed the data which includes only those cases exhibiting invasive breast cancer and no evidence of distant metastases at the time of diagnosis. The original data set has 198 observations (151 nonrecurrent and 47 recurrent) and 30 features which were computed from a digitized image of a fine needle aspirate of a breast mass. They describe characteristics of the cell nuclei present in the image. The mean, standard error, and "worst" or largest (mean of the three largest values) of these features have been computed for each image, resulting in a total of 30 features. However, some predictors are collinear or exhibit positive correlations beyond 0.99 which have been removed leaving a total of 29 predictors. Still, there are considerable correlations between the covariates which provides a challenging sampling problem.

| **short name** | **explanation** |
|---|---|
| TIME | recurrence time if recurrent, |
| | disease-free time if nonrecurrent |
| RADIUS | mean of distances from center |
| | to points on the perimeter |
| TEXTURE | standard deviation of gray-scale values |
| SMOOTHNESS | local variation in radius lengths |
| AREA | area |
| SMOOTHNESS | local variation in radius lengths |
| COMPACTNESS | perimeter$^2$ / area - 1.0 |
| CONCAVITY | severity of concave portions of the contour |
| CONCAVE_POINTS | number of concave portions of the contour |
| SYMMETRY | symmetry |
| FRACTAL_DIM | "coastline approximation" - 1 |
| TUMOR_SIZE | diameter of the excized tumor in centimeters |
| LYMPH_NODE | number of positive axillary lymph |
| | nodes observed at time of surgery |

**Musk data** The third example is based on a data set aiming at classifying whether a molecule is a muscle-specific kinase (MUSK) or not. Dietterich et al. (1997) use the original data to compare several axis-parallel rectangle algorithms. The dataset describes a set of 92 molecules of which 47 were judged by human experts to be MUSK and the

remaining 45 molecules were judged to be non-MUSK. The 166 features which describe the molecules depend upon the exact shape, or conformation, of the molecule. The total number of observations is 476. As in the Wisconsin Prognostic Breast Cancer example, some predictors are collinear or exhibit positive correlations beyond 0.99 which have been removed leaving a total of 95 predictors. The strong correlations between the covariates yield a challenging sampling problem.

| **short name** | **explanation** |
|---|---|
| DF_* | distance features |
| OXY_DIS | The distance of the oxygen atom in the molecule to a designated point in 3-space. |
| OXY_X | X-displacement from the designatet point. |
| OXY_Y | Y-displacement from the designated point. |
| OXY_Z | Z-displacement from the designated point. |

### 5.4.2. Comparison and conclusion

In this section, we provide a rough comparison between the pseudo-marginal SMC from Section 5.2.4, the corrected Laplace SMC from Section 5.2.5 and the automatic generic sampler from Section 5.3. In Section 4.5.2, we argued that for comparing completely different algorithms, pure computational time might not be the best criterion and preferred to calibrate the algorithms in terms of evaluations of the target function $\pi$. In the context of generalized linear models, we can hardly do the same since the automatic generic sampler works on the joint distribution and the adapted SMC samplers on the marginal distribution of the posterior. Therefore, we calibrate the pseudo-marginal SMC and the automatic generic sampler to have approximately the same running time. The corrected Laplace SMC approach proposed in Section 5.2.5 runs with the same configuration as the pseudo-marginal SMC but is significantly faster.

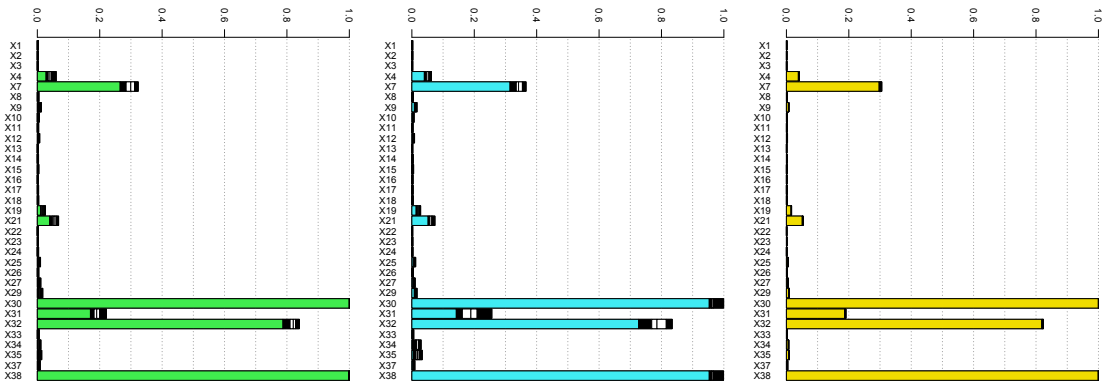
We run each algorithm 50 times and each time we obtain a Monte Carlo estimate of the marginal probabilities of inclusion of all predictors. We visualize the variation of the estimator by box-plots that show how much the Monte Carlo estimates have varied throughout the 50 runs (Figures 5.1 to 5.3). Here, the white boxes contain 80% of the Monte Carlo results, while the black boxes show the extent of the 20% outliers. For better readability, we add a colored bar up to the smallest estimate we obtained in the test runs; otherwise components with a small variation are hard to see. The vertical line in the white box indicates the median of the Monte Carlo estimates.

Clearly, on grounds of our comparison we cannot state that an SMC approach is better or worse than a transdimensional MCMC algorithm, since both methods may require a certain amount of problem-dependent tuning and good programming skills to be efficient. However, we may conclude that the pseudo-marginal SMC sampler is a viable alternative to transdimensional MCMC and produces results of similar accuracy for the same amount of computational time. The pseudo-marginal approach in a pure MCMC context would certainly not work as well, since many more evaluations of the target function were required.

Remember that the SMC sampler can compute the estimates of the marginal posterior in parallel and thus easily profit from parallel computing environments. Since computation of the marginals is the computationally most intensive step, even simple parallelization approaches lead to an enormous speed-up. We implemented a parallel version of the sampler, but only used a single core for the numerical comparison. We refer to the Appendix for details on the software.

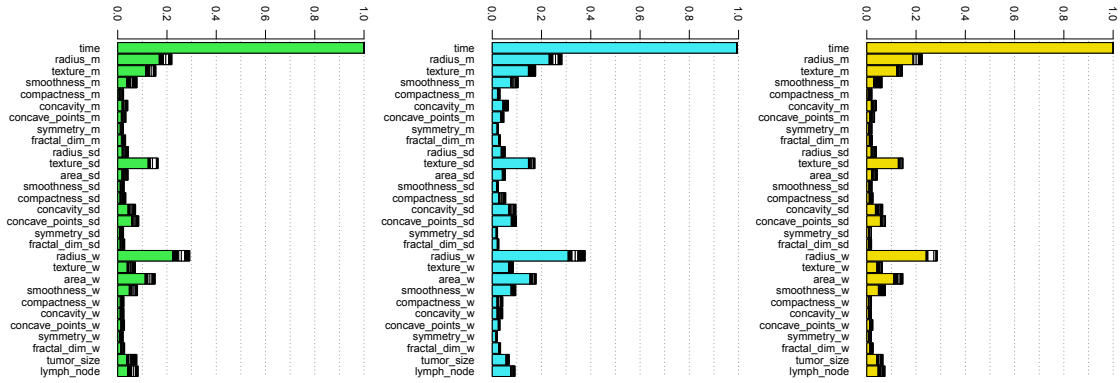
We also observe that the corrected Laplace approximation of the full posterior as proposed in Section 5.2.5 provides, from a practical point of view, a fast and rather reliable alternative to transdimensional MCMC. This sampling scheme puts us back into the SMC framework discussed in Chapter 2, where the target distribution is available in closed-form, but the sampler has to deal with multi-modality issues.

**Figure 5.1.:** Australian credit approval set. For details see Section 5.4.1. The average run time is about 16 minutes for the pseudo-marginal SMC and the automatic generic sampler.



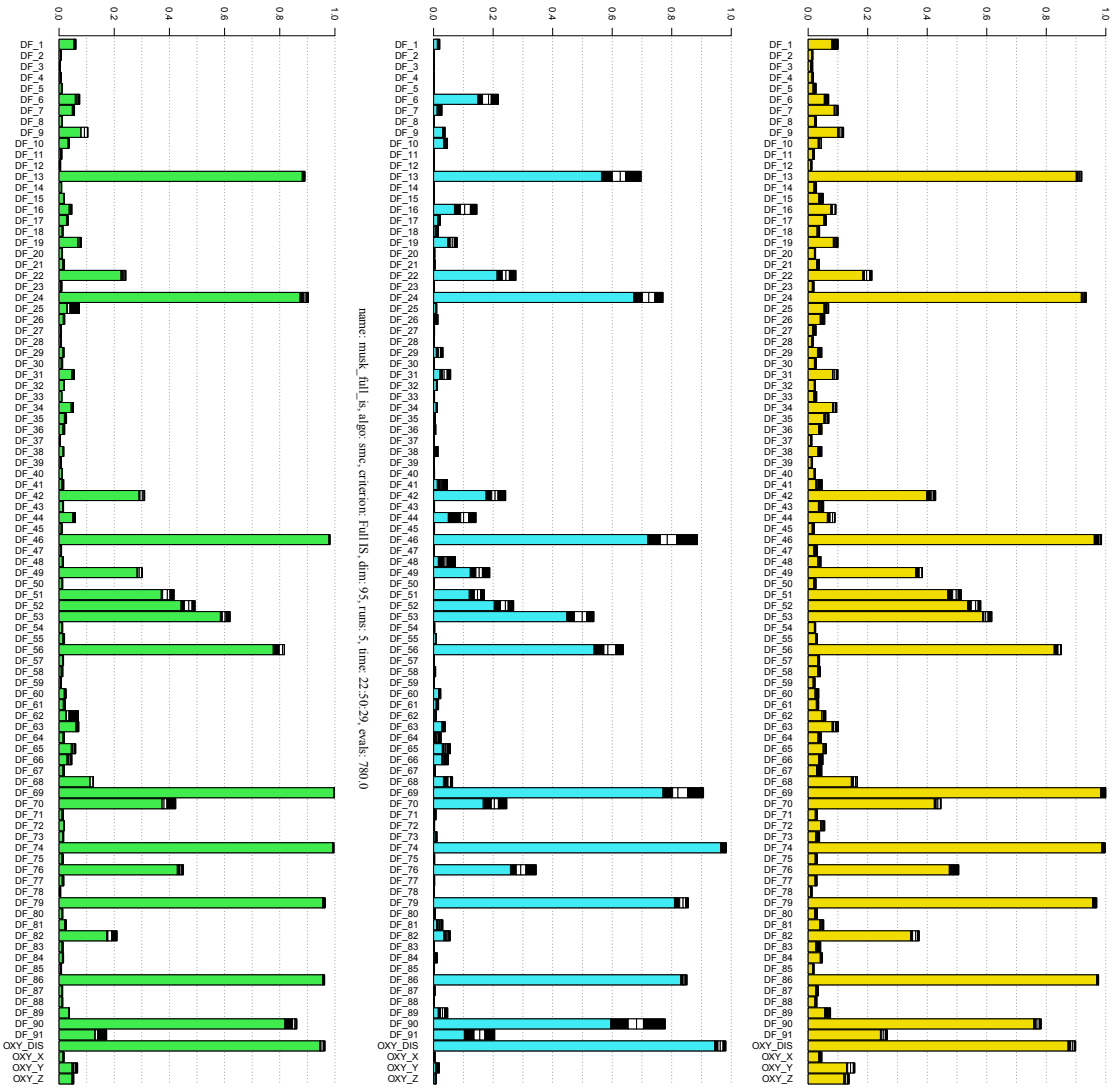
(a) pseudo-marginal SMC (b) automatic generic sampler (c) corrected Laplace SMC

**Figure 5.2.:** Wisconsin Prognostic Breast Cancer data set. For details see Section 5.4.1. The average run time is about 22 minutes for the pseudo-marginal SMC and the automatic generic sampler.




(a) pseudo-marginal SMC (b) automatic generic sampler (c) corrected Laplace SMC

**Figure 5.3.:** MUSK detection data set. For details see Section 5.4.1. The average run time is about 19 hours for the pseudo-marginal SMC and the automatic generic sampler.




(a) pseudo-marginal SMC (b) automatic generic sampler (c) corrected Laplace SMC

# 6. Pseudo-Boolean optimization

## Resumé

*L'optimisation stochastique de fonctions pseudo-booléennes est un domaine d'intérêt majeur en recherche opérationnelle car des nombreuses problèmes combinatoires NP-complet peuvent être formulés en termes de programmation binaire. Si la fonction objective est multimodale, les algorithmes de recherche locale ne parviennent souvent pas à détecter l'optimum global et les méthodes particulaires peuvent donner des résultats plus robustes. Nous détaillons comment l'échantillonneur de Monte Carlo séquentiel peut être utilisé dans un contexte d'optimisation et comment la méthode de l'entropie croisée par Rubinstein (1997) peut être intégré dans le cadre de l'algorithme Monte Carlo séquentiel. Les expériences numériques montrent que les familles paramétriques proposées dans le chapitre 3 améliorent considérablement la performance de la méthode de l'entropie croisée. Finalement, les méthodes particulaires sont comparées aux algorithmes de recherche locale.*

## 6.1. Introduction

We apply the sequential Monte Carlo (SMC) sampler developed in Chapter 2 to optimization problems. The material has been accepted for publication in Schäfer (2012b). In the context of combinatorial optimization, a mapping $f\colon \mathbb{B}^d \to \mathbb{R}$ is usually referred to as a *pseudo-Boolean function*. This terminology stems from the definition of a Boolean function $f\colon \mathbb{B}^d \to \mathbb{B}$ for logical calculation while the term *binary function* usually refers to functions with two input variables. In this chapter, we discuss a unified approach to stochastic optimization of pseudo-Boolean functions based on particle methods, including the cross-entropy method and simulated annealing as special cases.

We point out the need for auxiliary sampling distributions, that is parametric families on binary spaces, which are able to reproduce complex dependency structures, and illustrate their usefulness in our numerical experiments. We provide numerical evidence

that particle-driven optimization algorithms based on parametric families yield superior results on strongly multi-modal optimization problems while local search heuristics outperform them on easier problems.

In the following, we discuss approaches to obtain heuristics for the pseudo-Boolean optimization program

$$\begin{aligned} &\text{maximize} && f(\boldsymbol{x}) \\ &\text{subject to} && \boldsymbol{x} \in \mathbb{B}^d \end{aligned} \tag{6.1}$$

using SMC techniques, and we refer to $f$ as the *objective function*. Pseudo–Boolean optimization is equivalent to many combinatorial problems arising, for example, in reliability theory, design of integrated circuits, statistical mechanics, molecular conformation, operations research and management science, computer aided design, traffic management or machine scheduling. A large number of important combinatorial problems on graphs can be be formulated as optimization of quadratic pseudo-Boolean functions, including how to determine maximum vertex packings, maximum cliques, maximum cuts and minimum coverings. For an excellent overview of applications of binary programming and equivalent problems we refer to the survey paper by Boros and Hammer (2002) and references therein.

The idea to use particle filters for global optimization is not new (Del Moral et al., 2006, Section 2.3.1.c), but novel SMC methodology introduced in Chapter 2 allows to construct more efficient samplers for the special case of pseudo-Boolean optimization. We particularly discuss how this methodology connects with the cross-entropy method (Rubinstein, 1997), which is a well-established particle driven optimization algorithm based on parametric families. The SMC algorithm as developed in Chapter 2 is rather complex compared to local search algorithms such as simulated annealing (Kirkpatrick et al., 1983) or $k$-opt local search (Merz and Freisleben, 2002) which can be implemented in a few lines. The aim of this chapter is to motivate the use of particle methods in the context of pseudo-Boolean optimization and exemplify their usefulness on instances of the unconstrained quadratic binary optimization problem.

We investigate the performance of the proposed parametric families in particle-driven optimization algorithms and compare variants of the SMC algorithm, the cross-entropy method, simulated annealing and simple multiple-restart local search to analyze their respective efficiency in the presence or absence of strong local maxima. We provide conclusive numerical evidence that these complicated algorithms can indeed outperform simple heuristics if the objective function has poorly connected strong local maxima. This is not at all clear, since, in terms of computational time, multiple randomized restarts of fast local search heuristics might very well be more efficient than compara-

tively complex particle approaches.

### 6.1.1. Statistical modeling

For particle optimization, the common approach is defining a family of probability measures $(\pi_\varrho)_{\varrho\geq 0}$ associated to the optimization problem (6.1) in the sense that

$$\pi_0 = \mathcal{U}_{\mathbb{B}^d}, \quad \lim_{\varrho\to\infty} \pi_\varrho = \mathcal{U}_{M_f},$$

where $\mathcal{U}_S$ denotes the uniform distribution on the set $S$ and $M_f = \operatorname{argmax}_{\boldsymbol{x}\in\mathbb{B}^d} f(\boldsymbol{x})$ the set of maximizers. The idea behind this approach is to first sample from a simple distribution, potentially learn about the characteristics of the associated family and smoothly move towards distributions with more mass concentrated in the maxima. We review two well-known techniques to explicitly construct such a family $\pi_\varrho$.

**Definition 6.1.1.** We call $\{\pi_\varrho \colon \varrho \geq 0\}$ a tempered family, if it has probability mass functions of the form

$$\pi_\varrho(\boldsymbol{\gamma}) := \nu_\varrho \exp[\varrho\, f(\boldsymbol{\gamma})], \tag{6.2}$$

where $\nu_\varrho^{-1} := \sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} \exp[\varrho\, f(\boldsymbol{\gamma})]$.

As $\varrho$ increases, the modes of $\pi_\varrho$ become more accentuated until, in the limit, all mass is concentrated on the set of maximizers. The name reflects the physical interpretation of $\pi_\varrho(\boldsymbol{x})$ as the probability of a configuration $\boldsymbol{x} \in \mathbb{B}^d$ for an inverse temperature $\varrho$ and energy function $-f$. This is the sequence used in simulated annealing (Kirkpatrick et al., 1983).

**Definition 6.1.2.** We call $\{\pi_\varrho \colon \varrho \geq 0\}$ a level set family, if it has probability mass functions of the form

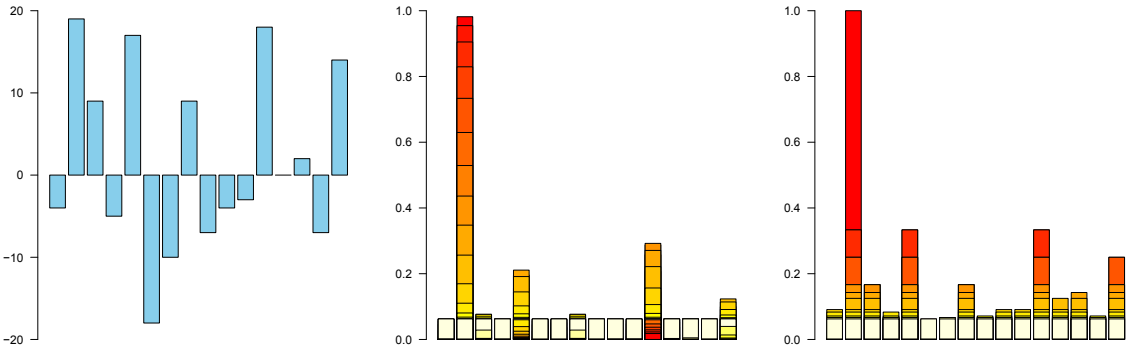
$$\pi_\varrho(\boldsymbol{\gamma}) := |L_\varrho^+|^{-1} \mathbb{1}_{L_\varrho^+}(\boldsymbol{\gamma}), \tag{6.3}$$

where $L_\varrho^+ := \{\boldsymbol{\gamma} \in \mathbb{B}^d \colon \varrho[f(\boldsymbol{x}^*) - f(\boldsymbol{\gamma})] \leq 1\}$ for $\boldsymbol{x}^* \in M_f$.

Indeed, $L_\varrho^+$ is the super-level set of $f$ with respect to the level $c = f(\boldsymbol{x}^*) - 1/\varrho$, for $\varrho > 0$, and $\pi_\varrho(\boldsymbol{\gamma})$ is the uniform distribution on $L_\varrho^+$. As $\varrho$ increases, the support of $\pi_\varrho$ becomes restricted to the points that have an objective value sufficiently close to the maximum of the $f$. In the limit, the support is reduced to the set of global maximizers.

Figure 6.1 shows a toy instance of an objective function on a discrete state space and two sequences associated to the optimization problem (6.1). The particle-driven optimization algorithms are computationally more involved than local search heuristics

**Figure 6.1.:** Associated sequences $\pi_{\varrho_t}$ for a toy example $f\colon \mathbb{B}^4 \to [-20, 20]$. The colors indicate the advance of the sequences from yellow to red. For simplicity, we choose $\varrho_t = t$ for $t \in [\![0, 16]\!]$.




(a) objective function $f(\boldsymbol{x})$ (b) tempered sequence (6.2) (c) level set sequence (6.3)

since we need to construct a sequence of distributions instead of a sequence of states. We shall see that this effort pays off in strongly multi-modal scenarios, where even sophisticated local search heuristics can get trapped in a subset of the state space.

### 6.1.2. Rare event simulation

While the tempered sequence is based on a physical intuition, the level set sequence has an immediate interpretation as a sequence of rare events since, as $\varrho$ increases, the super-level set becomes a 'rare event' with respect to the uniform measure. Rare event simulation and global optimization are therefore closely related concepts and methods for rare event estimation can often be adapted to serve as optimization algorithms.

Particle algorithms for rare event simulation include the cross-entropy method (Rubinstein, 1997) and the SMC sampler (Johansen et al., 2006). The former uses the level set sequence, the latter uses a *logistic potential family*

$$\pi_\varrho(\boldsymbol{\gamma}) := \nu_\varrho \, \ell(\varrho[f(\boldsymbol{\gamma}) - f(\boldsymbol{x}^*)]),$$

where $\nu_\varrho^{-1} := \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} \ell(\varrho[f(\boldsymbol{\gamma}) - f(\boldsymbol{x}^*)])$ and $\ell\colon \mathbb{R} \to (0,1)$, $\ell(x) = [1 + \exp(-x)]^{-1}$ denotes the logistic function. Johansen et al. (2006) did not specifically design their algorithm for optimization but their approach to static rare event simulation is closely related to the particle optimization framework.

## 6.2. Optimization algorithms

In this section, we briefly review some well-known heuristics for binary optimization. In particular, we discuss how the SMC algorithm introduced in Chapter 2 connects to the cross-entropy method and simulated annealing. In Table 6.1, we provide the necessary formulas for the tempered and the level set sequence introduced in Section 6.1.1.

**Table 6.1.:** Formulas of the importance function $u_{t,\alpha}$, the effective sample size $\eta_n$ and the acceptance probability $\lambda_q$ for the tempered and the rare event sequences.

| | $\exp(\varrho f)$ | $\mathbb{1}_{L_\varrho^+}$ |
|---|---|---|
| $u_{t,\alpha}(\boldsymbol{x}_{k,t})$ | $e^{\alpha f(\boldsymbol{x}_{k,t})}$ | $\mathbb{1}_{L_{\varrho_t+\alpha}^+}(\boldsymbol{x}_{k,t})$ |
| $\eta_n(\boldsymbol{w}_{t,\alpha})$ | $\frac{\left[\sum_{k=1}^n e^{\alpha f(\boldsymbol{x}_{k,t})}\right]^2}{\sum_{k=1}^n e^{2\alpha f(\boldsymbol{x}_{k,t})}}$ | $\lvert\{\boldsymbol{x}_{k,t} \mid k \in [\![1,n]\!]\} \cap L_{\varrho_t+\alpha}^+\rvert$ |
| $\lambda_{q_{t+1}}(\boldsymbol{\gamma} \mid \boldsymbol{x}_{k,t})$ | $1 \wedge \frac{e^{\alpha(f(\boldsymbol{\gamma})-f(\boldsymbol{x}_{k,t}))}}{e^{\log q_t(\boldsymbol{\gamma})-\log q_t(\boldsymbol{x}_{k,t})}}$ | $1 \wedge \frac{\mathbb{1}_{L_{\varrho_{t+1}}^+}(\boldsymbol{\gamma})}{e^{\log q_t(\boldsymbol{\gamma})-\log q_t(\boldsymbol{x}_{k,t})}}$ |

### 6.2.1. Sequential Monte Carlo

The SMC algorithm proceeds as described in Chapter 2 but does not terminate when $\varrho$ reaches exactly one. The iterations terminate if the particle diversity drops sharply below some threshold $\delta > 0$ which indicates that the mass has concentrated in a single mode. For convenience, the optimization scheme is summarized again in Algorithm 12.

If the Markov kernel is of the Metropolis-Hastings type with proposals from a parametric family $q_\theta$, one might already stop if the family degenerates in the sense that only a few components of $q_\theta$, say less than $d^* = 12$, are random while the others are constant ones or zeros. In this situation, additional moves using this parametric family are a pointless effort. We either return the maximizer within the particle system or we solve the subproblem of dimension $d^*$ by brute force enumeration. We might also perform some final local moves in order to further explore the regions of the state space the particles concentrated on.

For the level set sequence, the effective sample size is the fraction of the particles which have an objective function value greater than $\max_{\boldsymbol{\gamma}\in\mathbb{B}^d} f(\boldsymbol{\gamma}) - (\varrho+\alpha)^{-1}$, see Table

**Algorithm 12:** Sequential Monte Carlo optimization

**Input**: $f\colon \mathbb{B}^d \to \mathbb{R}$
**for all** $k \in N$ **sample** $\boldsymbol{x}_k \sim \mathcal{U}_{\mathbb{B}^d}$
**repeat**
  $\alpha \leftarrow$ **find step length**$(\varrho, \mathbf{X})$ (Procedure 4)
  $\boldsymbol{w} \leftarrow$ **importance weights**$(\alpha, \pi_\varrho, \mathbf{X})$ (Procedure 3)
  $\varrho \leftarrow \varrho + \alpha$
  $\theta \leftarrow$ **fit parametric family**$(\boldsymbol{w}, \mathbf{X})$ (see Chapter 3)
  $\widehat{\mathbf{X}} \leftarrow$ **resample**$(\boldsymbol{w}, \mathbf{X})$ (Procedure 5)
  $\mathbf{X} \leftarrow$ **move**$(\kappa_\theta, \widehat{\mathbf{X}})$ (Procedure 6)
**until** $\zeta_n(\mathbf{X}) < \delta$ **or** $q_\theta$ **degenerated**
**return** $\operatorname{argmax}_{\boldsymbol{\gamma} \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}} f(\boldsymbol{\gamma})$

6.1 and equation (6.3). The remaining particles are discarded since their weights equal zero. Thus, the weighting and resampling steps collapse to ordering the particles $\boldsymbol{x}_k$ according to their objective values $f(\boldsymbol{x}_k)$ and keeping the $n(1-\beta)$ particles with the highest objective values. Consequently, there is no need to explicitly compute $\alpha$ as a solution of (1.2).

### 6.2.2. Cross-entropy method

The cross-entropy method has been applied successfully to a variety of combinatorial optimization problems, some of which are equivalent to pseudo-Boolean optimization (Rubinstein and Kroese, 2004), and is closely related to the proposed SMC framework. Rubinstein (1997), who popularized the use of level set sequences in the context of the cross-entropy method, refers to $n(1-\beta)$ particles with the highest objective function values as the *elite sample*. Like in the SMC sampler, these particles are used to fit the next parameter of the auxiliary family.

However, the central difference between the cross-entropy method summarized in Algorithm 13 and the SMC algorithm outlined in Algorithm 12 is the use of an invariant transition kernel in the latter. We obtain the cross-entropy method as a special case of the SMC sampler if we replace the kernel $\kappa_\theta$ by its proposal distribution $q_\theta$.

The SMC annealing algorithm starts from a family of intermediate distributions $\{\pi_\varrho\colon \varrho \geq 0\}$ and explicitly schedules the evolution $(\pi_{\varrho_t})_{t\in\mathbb{N}}$ which in turn defines the proposal distributions $(q_{\theta_t})_{t\in\mathbb{N}}$. The cross-entropy method, in contrast, defines the subsequent proposal distribution

$$q_{\theta_{t+1}} \approx q_{\theta_t} \mathbb{1}_{L^+_{\varrho_{t+1}}}$$

without any reference sequence $(\pi_t)_{t\in\mathbb{N}}$ to balance the speed of the particle evolution.

In order to decelerate the advancement of the cross-entropy method, one might introduce a lag parameter $\tau \in [0, 1)$ and use a convex combination of the previous parameter $\theta_{t-1}$ and the parameter $\hat{\theta}_t$ fit to the current particle system, setting

$$\theta_t := (1 - \tau)\hat{\theta}_t + \tau\theta_{t-1}.$$

However, there are no guidelines on how to adjust the lag parameter during the run of the algorithm. Therefore, the SMC algorithm is easier to calibrate since the reference sequence $(\pi_t)_{t\in\mathbb{N}}$ controls the stride and automatically prevents the system from overshooting.

On the upside, the cross-entropy method allows for a broader class of auxiliary distributions $\{q_\theta \mid \theta \in \Theta\}$ since we do not need to evaluate $q_\theta$ point-wise which is only necessary for the computation of the Metropolis-Hastings ratio (1.7).

**Algorithm 13:** Cross-entropy method

**Input**: $f\colon \mathbb{B}^d \to \mathbb{R}$
**for all** $k \in N$ **sample** $\boldsymbol{x}_k \sim \mathcal{U}_{\mathbb{B}^d}$
**repeat**
  $\sigma \leftarrow$ **order such that** $\boldsymbol{x}_{\sigma(1)} \leq \cdots \leq \boldsymbol{x}_{\sigma(n)}$
  $\varrho \leftarrow f(x_{\sigma(\lfloor \beta n \rfloor)})$
  $\theta \leftarrow$ **fit parametric family**$(x_{\sigma(\lfloor \beta n \rfloor)}, \dots, x_{\sigma(n)})$ (see Section 3.1)
  **for all** $k \in N$ **sample** $\boldsymbol{x}_k \sim q_\theta$
**until** $\zeta_n(\mathbf{X}) < \delta$ **or** $q_\theta$ **degenerated**
**return** $\operatorname{argmax}_{\boldsymbol{\gamma}\in\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\}} f(\boldsymbol{\gamma})$

## 6.2.3. Simulated annealing

A well-studied approach to pseudo-Boolean optimization is simulated annealing (Kirkpatrick et al., 1983). While the name stems from the analogy to the annealing process in metallurgy, there is a pure statistical meaning to this setup. We can picture simulated annealing as approximating the mode of a tempered sequence (6.2) using a single particle. Since a single observation does not allow for fitting a parametric family, we have to rely on symmetric transition kernels (1.13) in the move step.

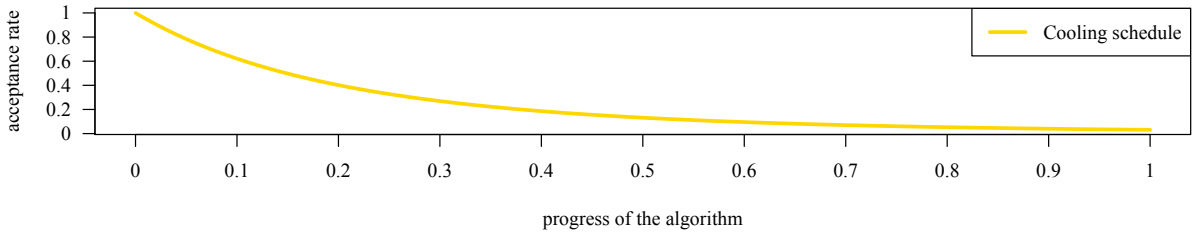
There is a vast literature advising on how to calibrate the sequence $(\varrho_t)_{t\in\mathbb{N}}$, which in this context is usually referred to as the *cooling schedule*, where a typical guideline is the

expected acceptance rate of the Hastings kernel. One might adaptively choose $(\varrho_t)_{t\in\mathbb{N}}$ such that the empirical acceptance rate

$$\overline{\alpha}_{t-s:t} := s^{-1} \sum_{r=t-s}^{t} \alpha_r$$

follows approximately a desired cooling schedule like $c\colon [0,\tau] \to [0,1]$, $c(t) = (1+\tau_\Delta/\tau)^{-5}$ where $\tau$ denotes the total running time and $\tau_\Delta$ the time elapsed while $s \in \mathbb{N}$ is some reasonable lag parameter. There are variants of simulated annealing which use more complex cooling schedules, tabu lists and multiple restarts, but we stick to this simple version for the sake of simplicity. Algorithm 14 describes the version we use in our numerical experiments in Section 6.4.4.

**Figure 6.2.:** The empirical acceptance probability is calibrated to follow $c(x) = (1+x)^{-5}$ where $x \in [0,\tau]$ is the progress of the simulated annealing algorithm.




**Algorithm 14:** Simulated annealing optimization

**Input**: $f\colon \mathbb{B}^d \to \mathbb{R}$, $\tau \in \mathbb{N}$
$\boldsymbol{x} \sim \mathcal{U}_{\mathbb{B}^d}$, $\boldsymbol{x}^* \leftarrow \boldsymbol{x}$, $t \leftarrow 0$, $\tau_\Delta \leftarrow 0$ (time elapsed)
**while** $t < \tau$ **do**
  **sample** $\boldsymbol{\gamma} \sim \mathcal{U}_{N_1(\boldsymbol{x})}$, $u \sim \mathcal{U}_{[0,1]}$
  **if** $u < \exp\left[\varrho\left(f(\boldsymbol{\gamma}) - f(\boldsymbol{x})\right)\right]$ **then** $\boldsymbol{x} \leftarrow \boldsymbol{\gamma}$
  **if** $f(\boldsymbol{x}) > f(\boldsymbol{x}^*)$ **then** $\boldsymbol{x}^* \leftarrow \boldsymbol{x}$
  **adjust** $\varrho$ **such that** $\overline{\alpha}_{t-s:t} \approx (1+\tau_\Delta/\tau)^{-5}$
  $t \leftarrow t+1$
**end**
**return** $\boldsymbol{x}^*$

### 6.2.4. Randomized local search

We describe a greedy local search algorithm which works on any state space that allows for defining a neighborhood structure. A greedy local search algorithm computes the objective value of all states in the current neighborhood and moves to the best state

found until a local optimum is reached. The local search algorithm is called $k$-opt if it searches the $k$-neighborhood defined in (1.9) (see e.g. Merz and Freisleben (2002) for a discussion).

The algorithm can be randomized by repeatedly restarting the procedure from randomly drawn starting points. There are more sophisticated versions of local search algorithms exploit the properties of the objective function but even a simple local search procedure can produce good results Alidaee et al. (2010). Algorithm 15 describes the 1-opt local search procedure we use in our numerical experiments in Section 6.4.4.

**Algorithm 15:** Randomized local search

**Input**: $f\colon \mathbb{B}^d \to \mathbb{R}$, $T^* \in \mathbb{R}$
$\boldsymbol{x}^* \sim \mathcal{U}_{\mathbb{B}^d}$, $T_\Delta \leftarrow 0$ (time elapsed)
**while** $T_\Delta < T^*$ **do**
  $\boldsymbol{x} \sim \mathcal{U}_{\mathbb{B}^d}$
  **while** $\boldsymbol{x}$ **is not a local optimum do**
    $\boldsymbol{x} \leftarrow \operatorname{argmax}_{\boldsymbol{\gamma} \in N_1(\boldsymbol{x})} f(\boldsymbol{\gamma})$
  **end**
  **if** $f(\boldsymbol{x}) > f(\boldsymbol{x}^*)$ **then** $\boldsymbol{x}^* \leftarrow \boldsymbol{x}$
**end**
**return** $\boldsymbol{x}^*$

## 6.3. Application

### 6.3.1. Unconstrained Quadratic Binary Optimization

Proposition 3.2.3 states that any pseudo-Boolean function $f\colon \mathbb{B}^d \to \mathbb{R}$ can be written as a multi-linear function

$$f(\boldsymbol{\gamma}) = \textstyle\sum_{I \subseteq D} a_I \prod_{i \in I} \gamma_i, \tag{6.4}$$

where $a_I \in \mathbb{R}$ are real-valued coefficients. We say the function $f$ is of order $k$ if the coefficients $a_I$ are zero for all $I \subseteq D$ with $|I| > k$. While optimizing a first order function is trivial, optimizing a non-convex second order function is already an NP-hard problem Garey and Johnson (1979).

In the sequel, we focus on optimization of second order pseudo-Boolean functions to exemplify the stochastic optimization schemes discussed in the preceding sections. If $f$ is a second order function, we rewrite program (6.1) as

$$\begin{aligned} &\text{maximize} && \boldsymbol{x}^\intercal \mathbf{F} \boldsymbol{x} \\ &\text{subject to} && \boldsymbol{x} \in \mathbb{B}^d, \end{aligned} \tag{6.5}$$

where $\mathbf{F} \in \mathbb{R}^{d \times d}$ is a symmetric matrix. The program (6.5) is called an unconstrained quadratic binary optimization (UQBO) problem; we refer to Boros et al. (2007) for a list of applications and equivalent problems. In the literature the problem is also denominated unconstrained quadratic Boolean or bivalent or zero-one programming (Beasley, 1998).

### 6.3.2. Particle optimization and meta-heuristics

Meta-heuristics are a class of algorithms that optimize a problem by improving a set of candidate solutions without systematically enumerating the state space; typically they deliver solutions in polynomial time while an exact solution has exponential worst case running time. The outcome is neither guaranteed to be optimal nor deterministic since most meta-heuristics are randomized algorithms. We briefly discuss the connection to particle optimization against the backdrop of the unconstrained quadratic binary optimization problem where we roughly separate them into two classes: local search algorithms and particle-driven meta-heuristics.

Local search algorithms iteratively improve the current candidate solution through local search heuristics and judicious exploration of the current neighborhood; examples are local search Boros et al. (2007); Merz and Freisleben (2002), tabu search Glover et al. (1998); Palubeckis (2004), simulated annealing Katayama and Narihisa (2001). Particle driven meta-heuristics propagate a set of candidate solutions and improve it through recombination and local moves of the particles; examples are genetic algorithms Merz and Freisleben (1999), memetic algorithms Merz and Katayama (2004), scatter search Amini et al. (1999). For comparisons of these methods we refer to Hasan et al. (2000) or Beasley (1998).

The SMC algorithm and the cross-entropy method are clearly in the latter class of particle-driven meta-heuristics. The idea behind SMC is closely related to the intuition behind population (or swarm) optimization and genetic (or evolutionary) algorithms. However, the mathematical framework used in SMC allows for a general formulation of the statistical properties of the particle evolution while genetic algorithms are often problem-specific and empirically motivated.

### 6.3.3. Particle optimization and exact solvers

If we can explicitly derive the multi-linear representation (6.4) of the objective function, there are techniques to turn program (6.1) into a linear program. For the UQBO it reads

$$\begin{aligned} \text{maximize} \quad & f(\boldsymbol{x}) = 2\sum_{i=1}^{d}\sum_{j=1}^{i-1} f_{ij}x_{ij} + \sum_{i=1}^{d} f_{ii}x_{ii} \\ \text{subject to} \quad & \boldsymbol{x} \in \mathbb{B}^{d(d+1)/2} \\ & \left.\begin{aligned} & x_{ij} \leq x_{ii} \\ & x_{ij} \leq x_{jj} \\ & x_{ij} \geq x_{ii} + x_{jj} - 1 \end{aligned}\right\} \text{ for all } i, j \in D. \end{aligned} \tag{6.6}$$

Note that there are more parsimonious linearization strategies than this straightforward approach (Hansen and Meyer, 2009; Gueye and Michelon, 2009). The transformed problem allows to access the tool box of linear integer programming which consist of branch-and-bound algorithms that are combined with rounding heuristics, various relaxations techniques and cutting plane methods (Pardalos and Rodgers, 1990; Palubeckis, 1995).

Naturally, the question arises whether particle-driven meta-heuristics can be incorporated into exact solvers to improve branch-and-bound algorithms. Indeed, stochastic meta-heuristics deliver lower bounds for maximization problems, but particle-driven algorithms are computationally somewhat expensive for this purpose unless the objective function is strongly multi-modal and other heuristics fail to provide good results; see the discussion in Section 6.3.4.

However, the SMC approach in combination with the level set sequence (6.3) might also be useful to determine a global branching strategy, since the algorithm provides an estimator for

$$\overline{\boldsymbol{\gamma}}_c := |L_c^+|^{-1} \textstyle\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} \boldsymbol{\gamma}\, \mathbb{1}_{L_c^+}(\boldsymbol{\gamma}),$$

which is the average of the super-level set $L_c^+ := \{\boldsymbol{x} \in \mathbb{B}^d \colon f(\boldsymbol{x}) \geq c\}$. These estimates given for a sequence of levels $c$ might provide branching strategies than are superior to local heuristics or branching rules based on fractional solutions. A further discussion of this topic is beyond the scope of this thesis but certainly merits consideration.

### 6.3.4. Construction of test problems

The meta-heuristics we want to compare do not exploit the quadratic structure of the objective function and might therefore be applied to any binary optimization program.

If the objective function can be written in multi-linear form like (6.5) there are efficient local search algorithms (Boros et al., 2007; Merz and Freisleben, 2002) which exploit special properties of the target function and easily beat particle methods in terms of computational time.

Therefore, the use of particle methods is particularly interesting if the objective function is expensive to compute or even a black box. The posterior distribution in Bayesian variable selection for linear normal models treated in Chapter 4 is an example of such an objective function. We stick to the UQBO for our numerical comparison since problem instances of varying difficulty are easy to generate and interpret while the results carry over to general binary optimization.

In the vast literature on UQBO, authors typically compare the performance of meta-heuristics on a suite of randomly generated problems with certain properties. Pardalos (1991) proposes standardized performance tests on symmetric matrices $\mathbf{F} \in \mathbb{Z}^{d\times d}$ with entries $f_{ij}$ drawn from the uniform

$$q_c(k) := \frac{1}{2c}\mathbb{1}_{[\![-c,c]\!]}(k), \quad c \in \mathbb{N}.$$

The test suites generated by Beasley (1990, OR-library) and Glover et al. (1998) follow this approach have been widely used as benchmark problems in the UQBO literature (see Boros et al. (2007) for an overview). In the sequel we discuss the impact of diagonal dominance, shifts, the density and extreme values of $\mathbf{F}$ on the expected difficulty of the corresponding UQBO problem.

### Diagonal

Generally, stronger diagonals in $\mathbf{F}$ corresponds to easier UQBO problems (Billionnet and Sutter, 1994). Consequently, the original problem generator presented by Pardalos (1991) is designed to draw the off-diagonal elements from a uniform on a different support $[\![-q, q]\!]$ with $q \in \mathbb{N}$.

The impact of the diagonal carries over to the statistical properties of the tempered distributions (6.2) defined in the introductory Section 6.1.1. For the UQBO, the tempered distributions are in the exponential quadratic family (3.5.1) and a strong diagonal implies low dependencies between the components of the random binary vector. Section 3.5.1 elaborates how to approximate the exponential quadratic family by the logistic conditionals family. One might accelerate the SMC algorithm using $p = q^{\ell}_{\mathbf{A}}$ instead of $p = \mathcal{U}_{\mathbb{B}^d}$ as initial distribution. However, we did not exploit this option to keep the present work more concise.

For positive definite $\mathbf{F}$, the optimization problem is convex and can be solved in polynomial time Kozlov et al. (1979); in exact optimization, this fact is exploited to construct upper bounds for maximization problems (Poljak and Wolkowicz, 1995). In statistical modeling, the auxiliary distribution

$$\pi(\boldsymbol{\gamma}) := \frac{\boldsymbol{\gamma}^\intercal \mathbf{F} \boldsymbol{\gamma}}{2^{d-2}\left(\mathbf{1}^\intercal \mathbf{F} \mathbf{1} + \operatorname{tr}(\mathbf{F})\right)},$$

is a feasible mass function for $\mathbf{F} > 0$. Section 3.4 provides analytical expressions for all cross-moments and marginal distributions without enumeration of the state space.

### Shifts

The global optimum of the UQBO problem is more difficult to detect as we shift the entries of the matrix $\mathbf{F}$ but the relative gap between the optimum and any heuristic value diminishes. If we sample $f_{ij} = f_{ij}^\tau$ from a uniform on the *shifted* support

$$q_{c,\tau}(k) := \mathcal{U}_{[\![-c+\tau, c+\tau]\!]}(k), \quad c \in \mathbb{N},\, \tau \in [\![-c, c]\!],$$

we obtain a random objective function

$$f_\tau(\boldsymbol{x}) = \boldsymbol{x}^\intercal \mathbf{F}^\tau \boldsymbol{x} \overset{d}{=} \boldsymbol{x}^\intercal (\mathbf{F}^0 + \tau \mathbf{1}\mathbf{1}^\intercal)\boldsymbol{x} = f_0(\boldsymbol{x}) + \tau \left|\boldsymbol{x}\right|^2,$$

where $\overset{d}{=}$ means equality in distribution. Hence, with growing $|\tau|$ the optimum depends less on $\mathbf{F}$ and the relative gap between the optimum and a solution provided by any meta-heuristic vanishes. Boros et al. (2007) define a related criterion for $\tau \in [\![-c, c]\!]$,

$$\bar{\rho} := \frac{1}{2} + \frac{\tau + 2\tau c}{2(\tau^2 + c^2 + c)} \in [0, 1],$$

and report a significant impact of $\bar{\rho}$ on the solution quality of their local search algorithms which is not surprising.

### Density

The difficulty of the optimization problem is related to the number of interactions, that is the number of non-zero elements of $\mathbf{F}$. We call the proportion of non-zeros the *density* of $\mathbf{F}$. Drawing $f_{ij}$ from the mixture

$$q_{c,\omega}(k) = \omega\, \mathcal{U}_{[\![-c,c]\!]}(k) + (1-\omega)\delta_0(k), \quad c \in \mathbb{N},\, \omega \in (0, 1]$$

we adjust the difficulty of the problem to a given expected density $\omega$.

Note that not all algorithms are equally sensitive to the density of $\mathbf{F}$. Using the basic linearization (6.6), each non-zero off-diagonal element requires the introduction of an auxiliary variable and three constraints. Thus, the expected total number of variables and the expected total number of constraints, which largely determine the complexity of the optimization problem, are proportional to the density $\omega$.

On the other hand, many randomized approaches, including the SMC sampler developed in Chapter 2, are less sensitive to the density of the problem in the sense that replacing zero elements by small values has a minor impact on the performance of these algorithms. Rather than the zero/non-zero duality, we suggest that the presence of extreme values determines the difficulty of providing heuristic solutions.

### Extreme values

The uniform sampling approach advocated by Pardalos (1991) is widely used in the literature for comparing meta-heuristics. Certainly, particle-driven methods are computationally too expensive to outperform local search heuristics on test problems with uniformly drawn entries; Beasley (1998) confirms this intuition with respect to genetic algorithms versus tabu search and simulated annealing. However, the uniform distribution does not produce *extreme values* and it is vital to keep in mind that these have an enormous impact on the performance of local search algorithms.

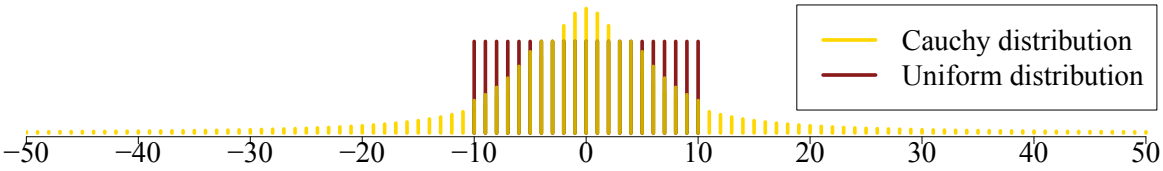
Extreme values in $\mathbf{F}$ lead to the existence of distinct local maxima $\boldsymbol{x}^* \in \mathbb{B}^d$ of $f$ in the sense that there is no better candidate solution than $\boldsymbol{x}^*$ in the neighborhood $H_k(\boldsymbol{x}^*)$ even for relatively large $k$. Further, extreme local minima might completely prevent a local search heuristic from traversing the state space in certain directions. Consequently, local search algorithms, as reviewed in Section 6.3.2, depend more heavily on their starting value, and their performance deteriorates with respect to particle-driven algorithms.

We propose to draw the matrix entries $f_{ij}$ from a discretized Cauchy distribution

$$\mathcal{C}_c(k) \propto (1 + (k/c)^2)^{-1}, \quad c \in \mathbb{N} \tag{6.7}$$

that has heavy tails which cause extreme values to be frequently sampled. Figure 6.3 shows the distribution of a Cauchy and a uniform to illustrate the difference. The resulting UQBO problems have quite distinct local maxima; in that case we also say that the function $f(\boldsymbol{x})$ is *strongly multi-modal*.

**Figure 6.3.:** Histograms of a Cauchy $\mathcal{C}_5$ and a uniform $\mathcal{U}_{10}$ distribution.




## 6.4. Numerical experiments

In this section, we provide numerical comparisons of algorithms and parametric families based on instances of the UQBO problem.

### 6.4.1. Toy example

We briefly discuss a toy example to illustrate the usefulness of the parametric families. For the quadratic function

$$f(\boldsymbol{x}) = \boldsymbol{x}^\intercal \mathbf{F} \boldsymbol{x}, \quad \mathbf{F} := \begin{pmatrix} 1 & 2 & 1 & 0 \\ 2 & 1 & -3 & -2 \\ 1 & -3 & 1 & 2 \\ 0 & -2 & 2 & -2 \end{pmatrix}, \tag{6.8}$$

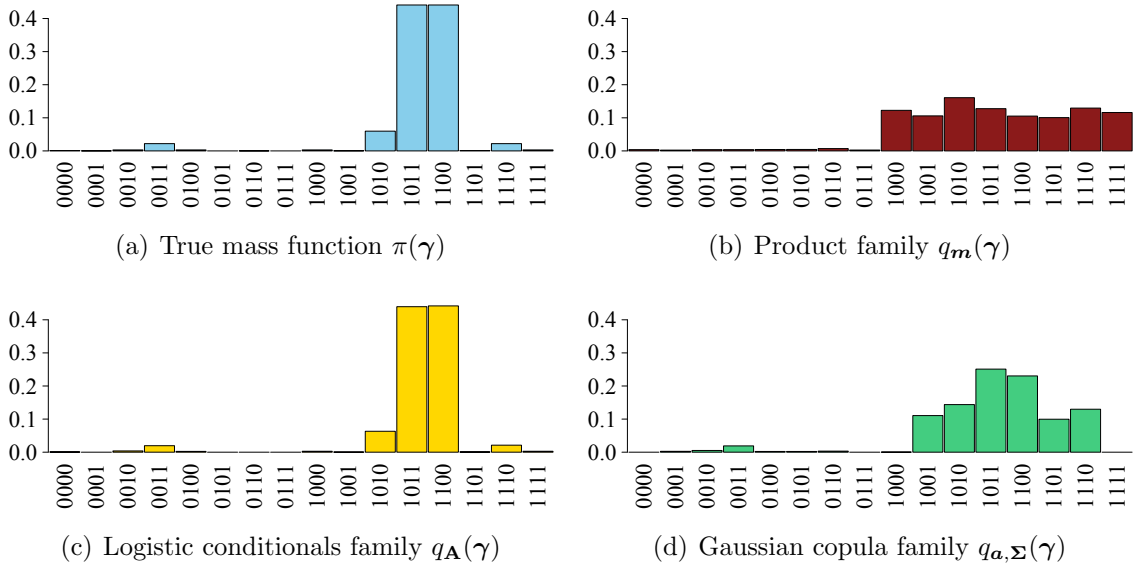
the associated probability mass function $\pi(\boldsymbol{\gamma}) \propto \exp(\boldsymbol{\gamma}^\intercal \mathbf{F} \boldsymbol{\gamma})$ has a correlation matrix

$$\mathbf{R} \approx \begin{pmatrix} 1 & 0.127 & -0.106 & -0.101 \\ 0.127 & 1 & -0.941 & -0.866 \\ -0.106 & -0.941 & 1 & 0.84 \\ -0.101 & -0.866 & 0.84 & 1 \end{pmatrix},$$

which indicates that this distribution has considerable dependencies and its mass function is therefore strongly multi-modal. We generate pseudo-random data from $\pi$, adjust the parametric families to the data and plot the mass functions of the fitted parametric families.

Figure 6.4 shows how the three parametric families cope with reproducing the true mass function. Clearly, the product family is not close enough to the true mass function to yield a suitable instrumental distribution while the logistic conditional family almost copies the characteristics of $\pi$ and the Gaussian copula family allows for an intermediate goodness of fit.

**Figure 6.4.:** Toy example showing how well the parametric families replicate the mass function of the distribution $\pi(\boldsymbol{\gamma}) \propto \exp(\boldsymbol{\gamma}^\intercal \mathbf{F} \boldsymbol{\gamma})$ as defined in (6.8).




(a) True mass function $\pi(\boldsymbol{\gamma})$

(b) Product family $q_{\boldsymbol{m}}(\boldsymbol{\gamma})$

(c) Logistic conditionals family $q_{\mathbf{A}}(\boldsymbol{\gamma})$

(d) Gaussian copula family $q_{\boldsymbol{a},\boldsymbol{\Sigma}}(\boldsymbol{\gamma})$

## 6.4.2. Random test instances

We generated two random test suites of dimension $d = 250$, each having 10 instances. For the first suite, we sampled the matrix entries uniformly on $[\![-100, 100]\!]$ that is from the distribution $\mathcal{U}_{100} := \mathcal{U}_{[\![-100,100]\!]}$; for the second, we sampled from a Cauchy distribution $\mathcal{C}_{100}$ as defined in (6.7). For performance evaluation, we run a specified algorithm 100 times on the same problem and denote the outcome by $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{100}$.

Since the absolute values are not meaningful, we report the relative ratios

$$\varrho_k := \frac{f(\boldsymbol{x}_k) - \text{worst solution found}}{\text{best known solution} - \text{worst solution found}} \in [0, 1],$$

where the best known solution is the highest objective value ever found for that instance and the worst solution is the lowest objective value among the 100 outcomes. We summarize the results in a histogram. The first $n$ bins are singletons $b_k := \{\varrho_k^*\}$ for the highest values $\varrho_1^* > \cdots > \varrho_n^* \in \{\varrho_k \colon k \in [\![1, 100]\!]\}$; the following $n$ bins are equidistant intervals $b_k^< := [\frac{n-k}{n}\varrho_n^*, \frac{n-k+1}{n}\varrho_n^*)$. The graphs show the bins $b_1, \ldots, b_n, b_1^<, \ldots, b_n^<$ in descending order from left to right on the $x$-axis. The interval bins are marked with a sign "$<$" and the lower bound. The $y$-axis represents the counts.

For comparison, we draw the outcome of several algorithms into the same histogram, where the worst solution found is the lowest overall objective value among the outcomes.

For each algorithm, the counts are depicted in a different color and, for better readability, with diagonal stripes in a different angle. To put it plainly, an algorithm performs well if its boxes are on the left of the graph since this implies that the outcomes where often close to the best known solution.

### 6.4.3. Comparison of binary parametric families

We study how the choice of the binary parametric family affects the quality of the delivered solutions. The focus is on the cross-entropy method, since we cannot easily use the Gaussian copula family in the context of SMC. For the experiments, we use $n = 1.2 \times 10^4$ particles, set the speed parameter to $\beta = 0.8$ (or the elite fraction to 0.2) and the lag parameter to $\tau = 0.5$.

The numerical comparisons, given in Figures 6.6(b) and 6.6(a), clearly suggest that using more advanced binary parametric families allows the cross-entropy method to detect local maxima that are superior to those detected using the product family. Hence, the numerical experiments confirm the intuition of our toy example in Figure 6.4.

On the strongly multi-modal instance 6.6(a) the numerical evidence for this conjecture is stunningly clear-cut; on the weakly multi-modal problem 6.6(b) its validity is still unquestionable. This result seems natural since reproducing the dependencies induced by the objective function is more relevant in the former case than in the latter.

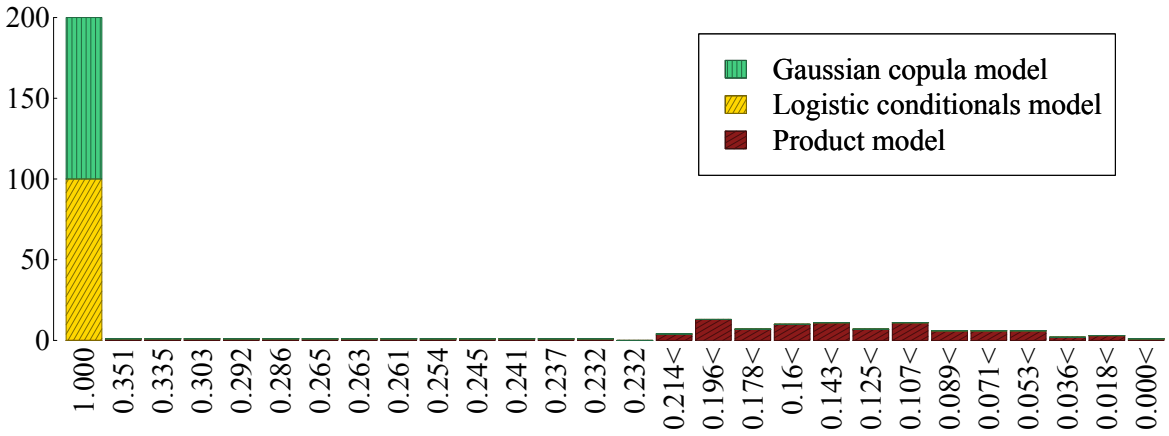
### 6.4.4. Comparison of optimization algorithms

We compare an SMC sampler with parametric family, an SMC sampler with single-flip symmetric kernel (1.13), the cross-entropy method, simulated annealing and 1-opt local search as described in Section 6.2.

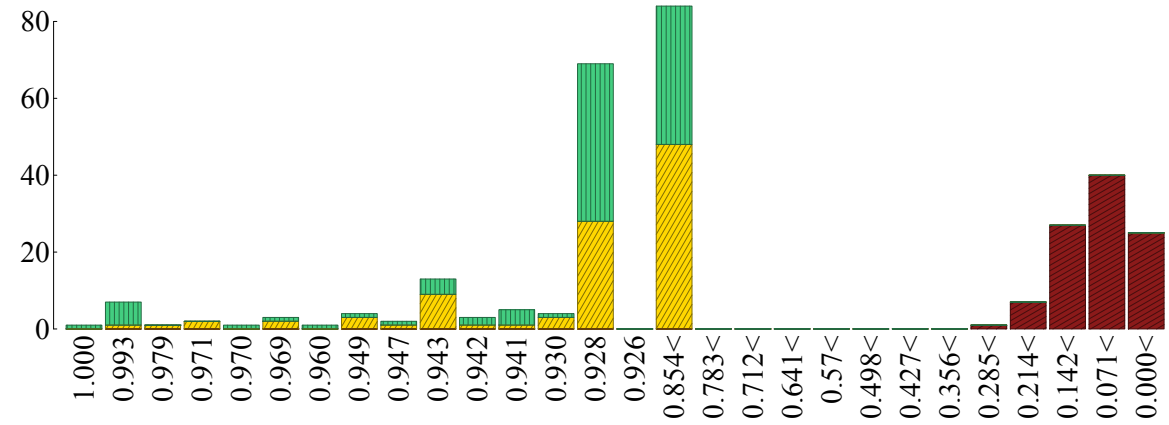
For the cross entropy method, we use the same parameters as in the preceding section. For the SMC algorithm, we use $n = 0.8 \times 10^4$ particles and set the speed parameter to $\beta = 0.9$; we target a tempered auxiliary sequence (6.2). For both algorithms we use the logistic conditionals family as sampling distribution. With these configurations, the algorithms converge in roughly 25 minutes. We calibrate the SMC sampler with local moves to have the same average run time by processing batches of 10 local moves before checking the particle diversity criterion. The simulated annealing and 1-opt local search algorithms run for exactly 25 minutes.

The results shown in Figures 6.7(b) and 6.7(a) assert the intuition that particle methods perform significantly better on strongly multi-modal problems. However, on

**Figure 6.5.:** The cross-entropy method using different binary parametric families.




(a) problem $f(\boldsymbol{x}) = \boldsymbol{x}^\intercal \mathbf{F} \boldsymbol{x}$ with $f_{ij} \sim \mathcal{C}_{100}$ for $i, j \in [\![1, 250]\!]$

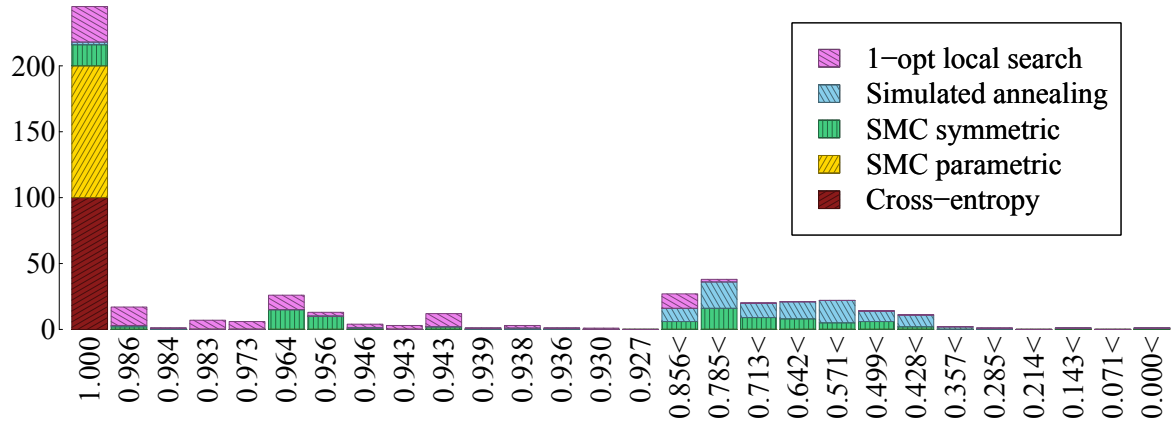



(b) problem $f(\boldsymbol{x}) = \boldsymbol{x}^\intercal \mathbf{F} \boldsymbol{x}$ with $f_{ij} \sim \mathcal{U}_{100}$ for $i, j \in [\![1, 250]\!]$

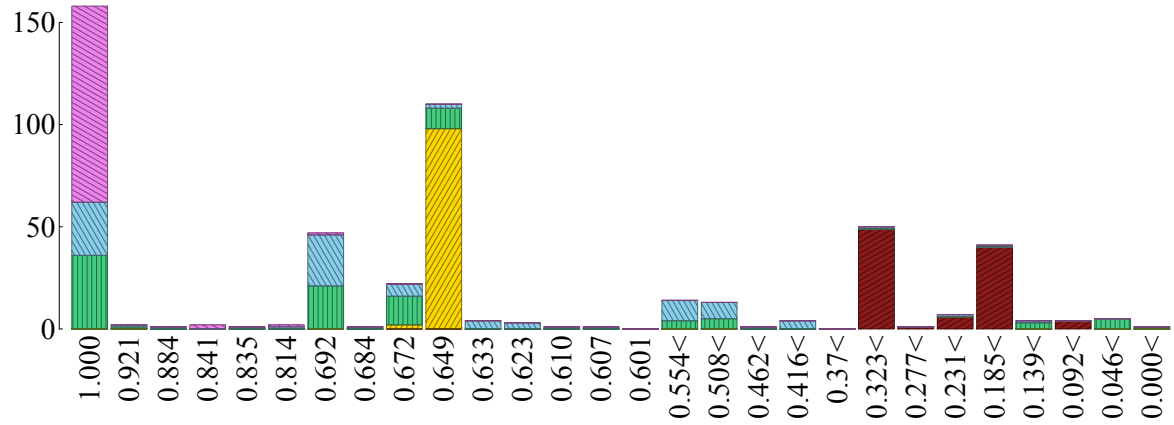
the easy test problems, the particle methods tend to persistently converge to the same sub-optimal local modes. This effect is probably due to their poor local exploration properties.

Since particle methods perform significantly less evaluations of the objective function, they are less likely to discover the highest peak in a region of rather flat local modes. The use of parametric families aggravates this effect, and it seems advisable to alternate global and local moves to make a particle algorithm more robust against this kind of behavior. Further numerical results are shown in Figure 6.7 and Figure 6.8.

**Figure 6.6.:** Comparison of stochastic optimization algorithms on two UQBO problems.




(a) problem $f(\boldsymbol{x}) = \boldsymbol{x}^\intercal \mathbf{F} \boldsymbol{x}$ with $f_{ij} \sim \mathcal{C}_{100}$ for $i, j \in [\![1, 250]\!]$



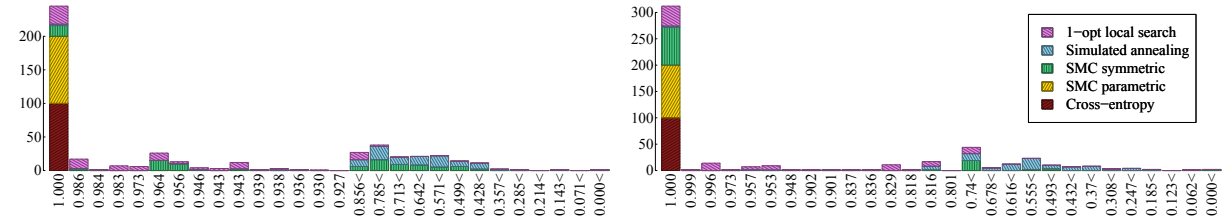

(b) problem $f(\boldsymbol{x}) = \boldsymbol{x}^\intercal \mathbf{F} \boldsymbol{x}$ with $f_{ij} \sim \mathcal{U}_{100}$ for $i, j \in [\![1, 250]\!]$

## 6.5. Discussion and conclusion

The numerical experiments carried out on different parametric families revealed that the use of the advanced families proposed in this paper significantly improves the performance of the particle algorithms, especially on the strongly multi-modal problems. The experiments demonstrate that local search algorithms, like simulated annealing and randomized 1-opt local search, indeed outperform particle methods on weakly multi-modal problems but deliver inferior results on strongly multi-modal problems.

Using tabu lists, adaptive restarts and rounding heuristics, we can certainly design local search algorithms that perform better than simulated annealing and 1-opt local search. Still, the structural problem of strong multi-modality persists for path-based algorithms. On the other hand, cleverly designed local search heuristics will clearly beat

**Figure 6.7.:** Comparison of stochastic optimization algorithms. 10 problems with objective function $f(\boldsymbol{x}) = \boldsymbol{x}^\intercal \mathbf{F} \boldsymbol{x}$ and $f_{ij} \sim \mathcal{C}_{100}$ for $i, j \in [\![1, 250]\!]$




(a) r250c problem 01

(b) r250c problem 02

(c) r250c problem 03

(d) r250c problem 04

(e) r250c problem 05

(f) r250c problem 06

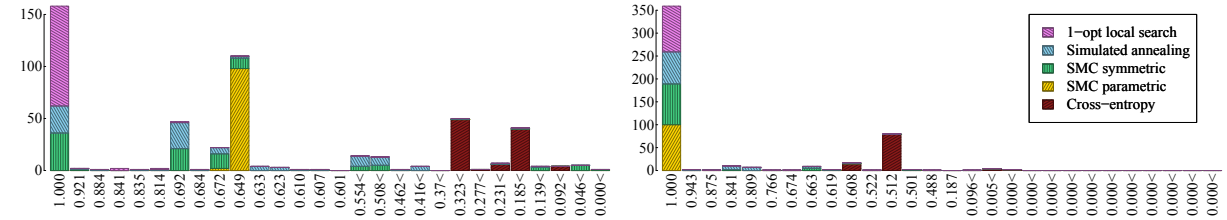
(g) r250c problem 07

(h) r250c problem 08

(i) r250c problem 09

(j) r250c problem 10

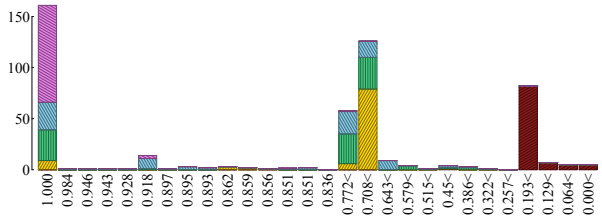
**Figure 6.8.:** Comparison of stochastic optimization algorithms. 10 problems with objective function $f(\boldsymbol{x}) = \boldsymbol{x}^\intercal \mathbf{F} \boldsymbol{x}$ and $f_{ij} \sim \mathcal{U}_{100}$ for $i, j \in [\![1, 250]\!]$



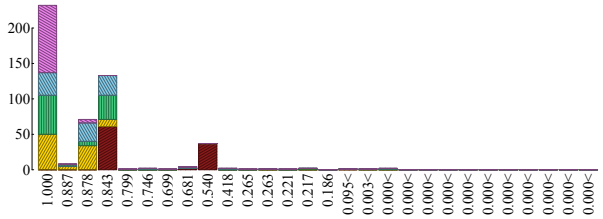
(a) r250u problem 01



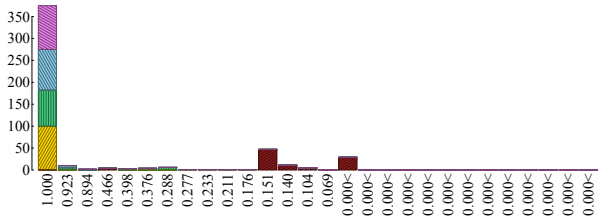

(b) r250u problem 02



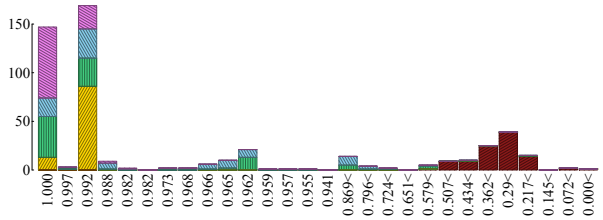
(c) r250u problem 03



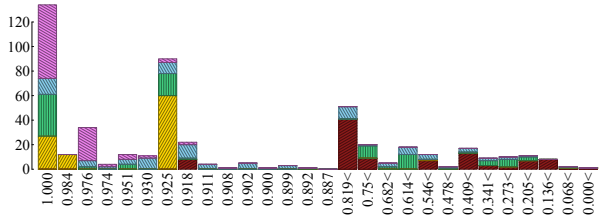
(d) r250u problem 04



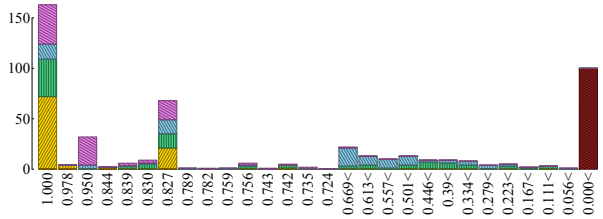
(e) r250u problem 05



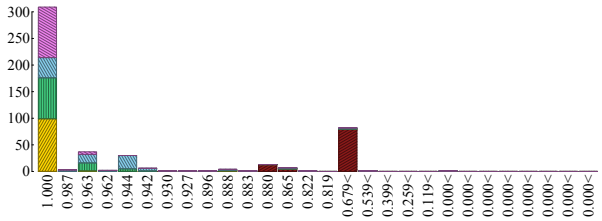
(f) r250u problem 06



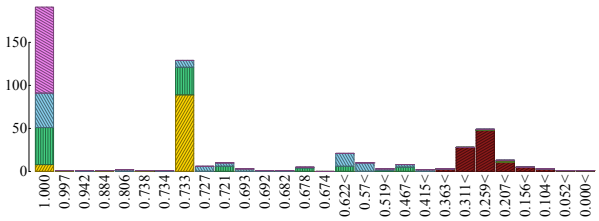
(g) r250u problem 07



(h) r250u problem 08



(i) r250u problem 09



(j) r250u problem 10

SMC methods on easy to moderately difficult problems.

The results encourage the use of particle methods if the objective function is known to be potentially multi-modal and hard to analyze analytically. We have to keep in mind that multiple restarts of rather simple local search heuristics can be very efficient if they make use of the structure of the objective function. For 25 minutes of randomized restarts, the heuristic proposed by Boros et al. (2007), which exploits the fact that the partial derivatives of a multi-linear function are constant, practically always returns the best known solution on all test problems treated to create Figures 6.7 and 6.8.

# 7. Conclusion and outlook

## Resumé

*La conclusion de cette thèse présente quelques remarques finales concernant les algorithmes particulaires sur les espaces d'états binaires et des perspectives de recherche pour intégrer les familles paramétriques dans d'autres applications.*

## 7.1. The independence sampler

The core work of this thesis is the thorough review of parametric families as a building block of adaptive Monte Carlo algorithms on binary sampling spaces. The sequential Monte Carlo (SMC) sampler with independent proposals based on these families has been shown to be rather robust when sampling from challenging multi-modal distributions of interest in the context of different applications. Admittedly, the implementation of the SMC sampler is rather involved compared to most Markov chain Monte Carlo (MCMC) methods, and this kind of methodology might be unnecessary on fairly easy sampling problems. Still, the SMC sampling scheme is very reliable, easy to tune and perfectly parallelizable.

The most important insight to be gained from this work is that a Metropolis-Hastings independence sampler with proposals drawn from an adaptive logistic conditionals family has excellent mixing properties and scales astonishingly well even to high dimensions. The "curse of dimensionality" which typically impedes the use of independent proposals does not seem to apply to binary spaces where we may construct parametric families to approximate even high-dimensional distributions of interest reasonably well.

The central problem is how to learn about the target distribution to be able to fit the parametric family. In this thesis we have proposed an annealing schedule in combination with an SMC sampler. However, there are other techniques coming from the tool box of adaptive Markov chain Monte Carlo (AMCMC) on binary spaces which may

also incorporate the Metropolis-Hastings independence sampler proposed in this thesis. This is particularly interesting since independent proposals allow for parallelization of the MCMC sampling scheme, see Section 7.3.

## 7.2. Scaling to higher dimensions

For testing, we also treated variable selection problems from association studies in plant genetics by courtesy of Willem Kruijer (Biometris Plant Sciences Group at Wageningen University) with 2000 predictors on a 64-CPU cluster using a parallelized version of the SMC sampler. The results were as reliable as for the test problem in Section 4.5 with about 100 predictors. These test runs are part of a comparison study for variable selection problems in the context of plant breeding which is on-going research. The results are still premature and therefore not included in this thesis.

The lesson to be learned from high-dimensional problems with more than 1000 predictors is that we do not need to work with an exponential number of particles just because the state space grows exponentially. In high dimensions, the reliability of the SMC sampling scheme can hardly be improved by using more particles but mostly depends on the number of resample-move steps we perform to stabilize the particle system. The central goal is to ensure that the particle system does not loose track of the intermediate distributions. This is obviously more difficult to achieve as the dimension of the sampling space increases and we need to choose the speed parameter $\eta^*$ introduced in equation (2.5) higher in order to follow the evolution of the intermediate distributions more closely. Generally, in high dimensions, the SMC estimator (2.1) is usually more efficient for the same amount of computational time if we use fewer particles but allow for more intermediate steps. This observation holds true for both Bayesian variable selection and pseudo-Boolean optimization.

## 7.3. Parallel computing

From a practical point of view, the possibility to parallelize the SMC sampler is even more interesting than its robustness against multi-modality when it comes to treat high-dimensional problems. Most researchers who process variable selection problems in applied fields have multi-core desktop computers, access to some kind of cluster or to a cloud computing service but there are few options to fully take advantage of these environments. The prototype implementation of the SMC sampler used for the numerical

studies in this thesis has shown the potential of our approach. Further improvements and better implementations of the SMC algorithm may shift the interest of practitioners towards particle methods for Bayesian variable selection.

The SMC sampler has the structural advantage that it may profit from as many cores as there are available in the computing environment. This is not true for random walk MCMC approaches. For example, parallel tempering algorithms obviously benefit from parallel computing, but there is a limit to the number of parallel chains which are useful to improve the mixing of the reference chain. If we have 8 CPUs we may run 8 parallel chains; if we have 256 CPUs available, we might still run 8 parallel chains if a finer temperature ladder does not improve the algorithm. However, in a pure AMCMC setup, the Metropolis-Hastings independence sampler based on the logistic conditionals family allows to fully benefit from parallel computing environments, since sampling proposals and evaluating the posterior mass function may be parallelized and sampling from the chain boils down to the Metropolis-Hastings acceptance step.

# Software

The numerical work in thesis was completely done in Python 2.6 using the SciPy package for scientific programming by Jones et al. (2001). Performance critical code was moved into C extensions written in Cython 0.14.1, a language which allows to tune Python code into plain C performance by adding static type declarations (Behnel et al., 2011). All graphs were generated using the R scripting language for statistical computing. The simulations were run on a 64 CPU cluster with 1.86 GHz processors.

The software and the variable selection problems processed in this thesis are made available along with some documentation at

`http://code.google.com/p/smcdss`.

The sequential Monte Carlo (SMC) and Markov chain Monte Carlo (MCMC) samplers for Bayesian variable selection are configured using an INI-file and may be run in a shell. There is support for automatic parallel computing on multiple CPUs based on the Parallel Python package by Vitalii Vanovschi.

For more convenient and self-explanatory use, we provide a simple graphical user interface written using the portable Tkinter module. The GUI allows to edit and organize the configuration files, monitor the performance of the samplers, create graphs in PDF format (calling R) and launch multiple external threads of the samplers.

# Glossary

| Notation | Description |
|---|---|
| $\boldsymbol{x} \in \mathbb{X}^d$ | vector of dimension $d$. |
| $\boldsymbol{x}_M \in \mathbb{X}^{\lvert M\rvert}$ | sub-vector indexed by $M \subseteq D$. |
| $\boldsymbol{x}_{i:j} \in \mathbb{X}^{j-i}$ | sub-vector indexed by $\{i, \dots, j\} \subseteq D$. |
| $\boldsymbol{x}_{-i} \in \mathbb{X}^{d-1}$ | sub-vector $\boldsymbol{x}_{D\setminus\{i\}}$. |
| $\lVert\boldsymbol{x}\rVert_\infty$ | $\max_{i\in D} \lvert x\rvert_i$. |
| $\lvert\boldsymbol{x}\rvert$ | $\sum_{i=1}^d \lvert x_i\rvert$. |
| $\lVert\pi\rVert_{\text{TV}}$ | $\frac{1}{2}\sum_{\boldsymbol{\gamma}\in\mathbb{B}^d} \lvert\pi(\boldsymbol{\gamma})\rvert$. |
| $f \propto g$ | $f = cg$ for some constant $c > 0$. |
| $x \vee y$ | Maximum. $x \vee y = \max\{x, y\}$. |
| $x \wedge y$ | Minimum. $x \wedge y = \min\{x, y\}$. |
| | |
| $\mathbf{A} = (a_{ij})$ | Matrix $\mathbf{A}$. |
| $\mathbf{A}^\intercal$ | Transpose of matrix $\mathbf{A}$. |
| $\mathbf{A}^{-1}$ | Inverse of matrix $\mathbf{A}$. |
| $\lvert\mathbf{A}\rvert$ | Determinant of $\mathbf{A}$. |
| $\text{diag}[\boldsymbol{a}]$ | Diagonal matrix with main diagonal $\boldsymbol{a}$. |
| $\lvert x\rvert$ | Absolute value of $x$. |
| $\mathbb{1}_M(x)$ | Indicator function of set $M$. |
| $\lvert M\rvert$ | Number of elements in the countable set $M$. |
| $\mathcal{P}(M)$ | Power set $\{S \subseteq M\}$. |
| $\mathcal{B}(M)$ | Borel $\sigma$-field $\{S \subseteq M \mid S \text{ is a Borel set}\}$. |
| $\text{supp}(f)$ | Support $\{f(x) \neq 0 \mid x \in \mathbb{X}\}$. |
| | |
| $\mathbb{B}$ | Binary space $\{0, 1\}$. |
| $\mathbb{N}$ | Set of natural numbers. |
| $\mathbb{Z}$ | Set of integer numbers. |
| $\mathbb{R}$ | Set of real numbers. |
| $[\![a, b]\!]$ | $\{x \in \mathbb{Z} \mid a \leq x \leq b\}$ for $a, b \in \mathbb{Z}$ with $b \geq a$. |
| $[a, b)$ | $\{x \in \mathbb{R} \mid a \leq x < b\}$ for $a, b \in \mathbb{R}$ with $b \geq a$. |

| **Notation** | **Description** |
|---|---|
| $D$ | Index set $[\![1, d]\!]$. |
| $N$ | Index set $[\![1, n]\!]$. |

# Acronyms

| Notation | Description |
|---|---|
| AIC | Akaike information criterion. |
| AMCMC | Adaptive Markov chain Monte Carlo. |
| BIC | Bayesian information criterion. |
| CE | Cross-entropy. |
| ESS | Effective sample size. |
| IID | Independent and indentically distributed. |
| IS | Importance sampling. |
| MAP | Maximum-a-posteriori. |
| MCMC | Markov chain Monte Carlo. |
| MUSK | Muscle-specific kinase. |
| SMC | Sequential Monte Carlo. |
| TV | Total variation. |
| UQBO | Unconstrained quadratic binary optimization. |

# Bibliography


Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Albert, A. and Anderson, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, 72:1–10.

Albert, J. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, pages 669–679.

Alidaee, B., Kochenberger, G., and Wang, H. (2010). Theorems supporting r-flip search for pseudo-Boolean optimization. *International Journal of Applied Metaheuristic Computing (IJAMC)*, 1(1):93–109.

Amini, M., Alidaee, B., and Kochenberger, G. (1999). A scatter search approach to unconstrained quadratic binary programs. In *New ideas in optimization*, pages 317–330. McGraw-Hill Ltd., UK.

Andrieu, C. and Roberts, G. (2009). The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics*, 37(2):697–725.

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373.

Arnold, B. (1996). Distributions with logistic marginals and/or conditionals. *Lecture Notes-Monograph Series*, 28:15–32.

Atchadé, Y. and Rosenthal, J. (2005). On adaptive Markov chain Monte Carlo algorithms. *Bernoulli*, 11(5):815–828.

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.

Barbieri, M. and Berger, J. (2004). Optimal predictive model selection. *The Annals of Statistics*, 32(3):870–897.

Beasley, J. (1990). OR-Library: Distributing test problems by electronic mail. *Journal of the Operational Research Society*, pages 1069–1072.

Beasley, J. (1998). Heuristic algorithms for the unconstrained binary quadratic programming problem. Technical report, Management School, Imperial College London.

Beaumont, M. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160.

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D., and Smith, K. (2011). Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31–39.

Billionnet, A. and Sutter, A. (1994). Minimization of a quadratic pseudo-Boolean function. *European Journal of Operational Research*, 78(1):106–115.

Bishop, Y., Fienberg, S., and Holland, P. (1975). *Discrete multivariate analysis: Theory and Practice*. Cambridge, MA: MIT Press.

Boros, E. and Hammer, P. (2002). Pseudo-Boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225.

Boros, E., Hammer, P., and Tavares, G. (2007). Local search heuristics for quadratic unconstrained binary optimization (QUBO). *Journal of Heuristics*, 13(2):99–132.

Bottolo, L. and Richardson, S. (2010). Evolutionary stochastic search for Bayesian model exploration. *Bayesian Analysis*, 5(3):583–618.

Brooks, S., Giudici, P., and Roberts, G. (2003). Efficient construction of reversible jump markov chain monte carlo proposal distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):3–39.

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.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved Particle Filter for nonlinear problems. *IEE Proc. Radar, Sonar Navigation*, 146(1):2–7.

Celeux, G., Anbari, M., Marin, J., and Robert, C. (2011). Regularization in regression: comparing bayesian and frequentist methods in a poorly informative situation. *arxiv preprint arXiv:1010.0300*.

Chaganty, N. and Joe, H. (2006). Range of correlation matrices for dependent Bernoulli random variables. *Biometrika*, 93(1):197–206.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, 89(3):539.

Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411.

Chopin, N., Jacob, P., and Papaspiliopoulos, O. (2011). Smcˆ 2: A sequential monte carlo algorithm with particle markov chain monte carlo updates. *Arxiv preprint arXiv:1101.1528*.

Clyde, M., Ghosh, J., and Littman, M. (2011). Bayesian adaptive sampling for variable selection and model averaging. *Journal of Computational and Graphical Statistics*, 20(1):80–101.

Clyde, M. and Parmigiani, G. (1998). Protein construct storage: Bayesian variable selection and prediction with mixtures. *Journal of biopharmaceutical statistics*, 8(3):431.

Cox, D. (1972). The analysis of multivariate binary data. *Applied Statistics*, pages 113–120.

Cox, D. and Wermuth, N. (1994). A note on the quadratic exponential binary distribution. *Biometrika*, 81(2):403–408.

Cox, D. and Wermuth, N. (1996). *Multivariate dependencies: Models, analysis and interpretation*, volume 67. Chapman & Hall/CRC.

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.

Del Moral, P., Doucet, A., and Jasra, A. (2012). An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, to appear. doi: 10.1007/s11222-011-9271-y.

Deville, J. and Tillé, Y. (2004). Efficient balanced sampling: the cube method. *Biometrika*, 91(4):893–912.

Diaconis, P. and Hanlon, P. (1992). Eigen analysis for some examples of the metropolis algorithm. *Contemporary Mathematics*, 138:99–117.

Dietterich, T., Lathrop, R., and Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71.

Dongarra, J., Moler, C., Bunch, J., and Stewart, G. (1979). *LINPACK: users' guide*. Society for Industrial and Applied Mathematics.

Douc, R., Cappé, O., and Moulines, E. (2005). Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. Ieee.

Drezner, Z. and Wesolowsky, G. O. (1990). On the computation of the bivariate normal integral. *Journal of Statistical Computation and Simulation*, 35:101–107.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of statistics*, 32(2):407–499.

Emrich, L. and Piedmonte, M. (1991). A method for generating high-dimensional multivariate binary variates. *The American Statistician*, 45:302–304.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Farrell, P. and Rogers-Stewart, K. (2008). Methods for generating longitudinally correlated binary data. *International Statistical Review*, 76(1):28–38.

Farrell, P. and Sutradhar, B. (2006). A non-linear conditional probability model for generating correlated binary data. *Statistics & probability letters*, 76(4):353–361.

Fearnhead, P. and Clifford, P. (2003). Online inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(4):887–899.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, 80:27–38.

Foster, D. and George, E. (1994). The risk inflation criterion for multiple regression. *The Annals of Statistics*, pages 1947–1975.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

Gange, S. (1995). Generating Multivariate Categorical Variates Using the Iterative Proportional Fitting Algorithm. *The American Statistician*, 49(2).

García-Donato, G. and Martínez-Beneito, M. (2011). Inferences in Bayesian variable selection problems with large model spaces. Technical report, arXiv:1101.4368v1.

Garey, M. and Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman & Co.

Gelman, A. and Meng, X. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13(2):163–185.

Genest, C. and Neslehova, J. (2007). A primer on copulas for count data. *Astin Bulletin*, 37(2):475.

Genz, A. and Bretz, F. (2002). Comparison of methods for the computation of multivariate t probabilities. *Journal of Computational and Graphical Statistics*, 11(4):950–971.

Genz, A. and Bretz, F. (2009). *Computation of multivariate normal and t probabilities*, volume 195. Springer.

George, E. I. and McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistica Sinica*, 7:339–373.

Ghosh, J. and Clyde, M. (2011). Rao–blackwellization for bayesian variable selection and model averaging in linear and binary regression: A novel data augmentation approach. *Journal of the American Statistical Association*, 106(495):1041–1052.

Gilks, W. and Berzuini, C. (2001). Following a moving target — Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146.

Glover, F., Kochenberger, G., and Alidaee, B. (1998). Adaptive memory tabu search for binary quadratic programs. *Management Science*, 44:336–345.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. Radar, Sonar Navigation*, 140(2):107–113.

Green, P. (2003). Trans-dimensional markov chain monte carlo. *Oxford Statistical Science Series*, pages 179–198.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Gueye, S. and Michelon, P. (2009). A linearization framework for unconstrained quadratic (0-1) problems. *Discrete Applied Mathematics*, 157(6):1255–1266.

Haberman, S. (1972). Algorithm AS 51: Log-linear fit for contingency tables. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 21(2):218–225.

Hamze, F., Wang, Z., and de Freitas, N. (2011). Self-Avoiding Random Dynamics on Integer Complex Systems. Technical report, arXiv:1111.5379.

Hans, C., Dobra, A., and West, M. (2007). Shotgun stochastic search for "large p" regression. *Journal of the American Statistical Association*, 102(478):507–516.

Hansen, M. and Yu, B. (2001). Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96(454):746–774.

Hansen, P. and Meyer, C. (2009). Improved compact linearizations for the unconstrained quadratic 0-1 minimization problem. *Discrete Applied Mathematics*, 157(6):1267–1290.

Harrison, D. and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102.

Hasan, M., AlKhamis, T., and Ali, J. (2000). A comparison between simulated annealing, genetic algorithm and tabu search methods for the unconstrained quadratic Pseudo-Boolean function. *Computers & Industrial Engineering*, 38(3):323–340.

Higdon, D. (1998). Auxiliary variable methods for markov chain monte carlo with applications. *Journal of the American Statistical Association*, pages 585–595.

Higham, N. J. (2002). Computing the nearest correlation matrix — a problem from finance. *IMA Journal of Numerical Analysis*, 22:329–343.

Hoeting, J., Madigan, D., Raftery, A., and Volinsky, C. (1999). Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401.

Holmes, C. and Held, L. (2006). Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1(1):145–168.

Jasra, A., Stephens, D., Doucet, A., and Tsagaris, T. (2011). Inference for Lévy-Driven Stochastic Volatility Models via Adaptive Sequential Monte Carlo. *Scandinavian Journal of Statistics*.

Johansen, A., Moral, P. D., and Doucet, A. (2006). Sequential Monte Carlo samplers for rare events. In *Proceedings of the 6th International Workshop on Rare Event Simulation*, pages pp. 256–267.

Johnson, N., Kotz, S., and Balakrishnan, N. (2002). *Continuous multivariate distributions - models and applications*, volume 2. New York: John Wiley & Sons,.

Jones, E., Oliphant, T., and Peterson, P. (2001). SciPy: Open source scientific tools for Python.

Kapur, J. (1989). *Maximum-entropy models in science and engineering*. John Wiley & Sons.

Kass, R. and Wasserman, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association*, pages 1343–1370.

Katayama, K. and Narihisa, H. (2001). Performance of Simulated Annealing-based heuristic for the unconstrained binary quadratic programming problem. *E. J. of Operational Research*, 134(1):103–119.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220(4598):671.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputation and Bayesian missing data problems. *Journal of the American Statistical Association*, 89:278–288.

Kosmidis, I. and Firth, D. (2009). Bias reduction in exponential family nonlinear models. *Biometrika*, 96(4):793–804.

Kozlov, M., Tarasov, S., and Khachiyan, L. (1979). Polynomial solvability of convex quadratic programming. In *Soviet Mathematics Doklady*, volume 20, pages 1108–1111.

Lamnisos, D., Griffin, J., and Steel, M. (2009). Transdimensional sampling algorithms for bayesian variable selection in classification problems with many more variables than observations. *Journal of Computational and Graphical Statistics*, 18(3):592–612.

Lamnisos, D., Griffin, J., and Steel, M. (2011). Adaptive monte carlo for bayesian variable selection in regression models.

Lee, A. (1993). Generating Random Binary Deviates Having Fixed Marginal Distributions and Specified Degrees of Association. *The American Statistician*, 47(3).

Lee, A., Yau, C., Giles, M., Doucet, A., and Holmes, C. (2010). On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *Journal of Computational and Graphical Statistics*, 19(4):769–789.

Liang, F., Paulo, R., Molina, G., Clyde, M., and Berger, J. (2008). Mixtures of g priors for Bayesian variable selection. *Journal of the American Statistical Association*, 103(481):410–423.

Liu, J. (1996a). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6(2):113–119.

Liu, J. (1996b). Peskun's theorem and a modified discrete-state Gibbs sampler. *Biometrika*, 83(3):681–682.

Liu, J. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044.

Lunn, A. and Davies, S. (1998). A note on generating correlated binary variables. *Biometrika*, 85(2):487–490.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*. Chapman & Hall / CRC, London.

Merz, P. and Freisleben, B. (1999). Genetic algorithms for binary quadratic programming. In *Proceedings of the genetic and evolutionary computation conference*, volume 1, pages 417–424. Citeseer.

Merz, P. and Freisleben, B. (2002). Greedy and local search heuristics for unconstrained binary quadratic programming. *Journal of Heuristics*, 8(2):197–213.

Merz, P. and Katayama, K. (2004). Memetic algorithms for the unconstrained binary quadratic programming problem. *BioSystems*, 78(1-3):99–118.

Meyn, S., Tweedie, R., and Glynn, P. (2009). *Markov chains and stochastic stability*, volume 2. Cambridge University Press Cambridge.

Modarres, R. (2011). High dimensional generation of bernoulli random vectors. *Statistics & Probability Letters*.

Neal, R. (2001). Annealed importance sampling. *Statistics and Computing*, 11(2):125–139.

Nelsen, R. (2006). *An introduction to copulas*. Springer Verlag.

Nott, D. and Green, P. (2004). Bayesian variable selection and the swendsen-wang algorithm. *Journal of computational and Graphical Statistics*, 13(1):141–157.

Nott, D. and Kohn, R. (2005). Adaptive sampling for Bayesian variable selection. *Biometrika*, 92(4):747.

Oman, S. and Zucker, D. (2001). Modelling and generating correlated binary variables. *Biometrika*, 88(1):287.

Palubeckis, G. (1995). A heuristic-based branch and bound algorithm for unconstrained quadratic zero-one programming. *Computing*, 54(4):283–301.

Palubeckis, G. (2004). Multistart tabu search strategies for the unconstrained binary quadratic optimization problem. *Annals of Operations Research*, 131(1):259–282.

Pardalos, P. (1991). Construction of test problems in quadratic bivalent programming. *ACM Transactions on Mathematical Software (TOMS)*, 17(1):74–87.

Pardalos, P. and Rodgers, G. (1990). Computational aspects of a branch and bound algorithm for quadratic zero-one programming. *Computing*, 45(2):131–144.

Park, C., Park, T., and Shin, D. (1996). A simple method for generating correlated binary variates. *The American Statistician*, 50(4).

Plackett, R. (1965). A class of bivariate distributions. *Journal of the American Statistical Association*, pages 516–522.

Poljak, S. and Wolkowicz, H. (1995). Convex relaxations of (0, 1)-quadratic programming. *Mathematics of Operations Research*, pages 550–561.

Qaqish, B. (2003). A family of multivariate binary distributions for simulating correlated binary variables with specified marginal means and correlations. *Biometrika*, 90(2):455.

Quinlan, J. (1987). Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234.

Robert, C. and Casella, G. (2004). *Monte Carlo statistical methods*. Springer Verlag.

Roberts, G. and Rosenthal, J. (2007). Coupling and ergodicity of adaptive markov chain monte carlo algorithms. *Journal of applied probability*, 44(2):458–475.

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.

Schäfer, C. (2012a). On parametric families for sampling binary data with specified mean and correlation. Technical report. arXiv:1111.0574.

Schäfer, C. (2012b). Particle algorithms for optimization on binary spaces. *ACM Transactions on Modeling and Computer Simulation*, (accepted for publication). arXiv:1111.0574.

Schäfer, C. and Chopin, N. (2012). Sequential Monte Carlo on large binary sampling spaces. *Statistics and Computing*, to appear. doi: 10.1007/s11222–011–9299–z.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, (6):461–464.

Soofi, E. (1994). Capturing the Intangible Concept of Information. *Journal of the American Statistical Association*, 89:1243–54.

Streitberg, B. (1990). Lancaster interactions revisited. *The Annals of Statistics*, 18(4):1878–1885.

Suchard, M., Holmes, C., and West, M. (2010). Some of the What?, Why?, How?, Who? and Where? of Graphics Processing Unit Computing for Bayesian Analysis. In et al. Oxford University Press. Bernardo, J. M., editor, *Bayesian Statistics 9*.

Swendsen, R. and Wang, J. (1987). Nonuniversal critical dynamics in monte carlo simulations. *Physical Review Letters*, 58(2):86.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Walker, A. (1977). An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software*, 3(3):256.

Wermuth, N. (1976). Analogies between multiplicative models in contingency tables and covariance selection. *Biometrics*, pages 95–108.

Wolberg, W., Street, W., and Mangasarian, O. (1995). Image analysis and machine learning applied to breast cancer diagnosis and prognosis. *Analytical and Quantitative cytology and histology*, 17(2):77–87.

Yeh, I. (1998). Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808.

Zellner, A. and Siow, A. (1980). Posterior odds ratios for selected regression hypotheses. *Trabajos de estadística y de investigación operativa*, 31(1):585–603.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.