Année: 2012

THÈSE

présentée à

L'UNIVERSITÉ PARIS–DAUPHINE
ÉCOLE DOCTORALE DE DAUPHINE
CEntre de **RE**cherche en **MA**thématiques de la **DE**cision

pour l'obtention du titre de

DOCTEUR EN MATHÉMATIQUES APPLIQUÉES

présentée et soutenue par
Pierre E. JACOB
le 3 septembre 2012

Contributions computationnelles à la statistique Bayésienne

Jury

Christian P. ROBERT	Université Paris–Dauphine	Directeur de thèse
Christophe ANDRIEU	University of Bristol	Rapporteurs
Nando DE FREITAS	University of British Columbia	
Nicolas CHOPIN	CREST–ENSAE	Examinateurs
Jean-Michel MARIN	Université Montpellier II	
Robin RYDER	Université Paris–Dauphine	

Remerciements

Ma plus profonde gratitude à Christian Robert, un vrai mentor. C'était un grand honneur de travailler sous sa direction attentive et bienveillante: il m'a guidé quand j'en avais besoin tout en respectant et en développant mon envie d'autonomie. Merci également pour son encouragement sans faille à voyager et à rencontrer du monde; ces périples ont changé beaucoup de choses, pour le mieux.

Je remercie le Fonds AXA pour la Recherche pour le financement de cette thèse et pour leur confiance. Merci également pour l'organisation des très sympathiques journées "Popularizing your Research". Merci également à l'administration de Dauphine pour leur efficacité, notamment Gilles Bares, Marie Belle, Isabelle Bellier, César Faivre et Michel Vanbreugel.

Merci aux rapporteurs Christophe Andrieu et Nando de Freitas pour leur lecture attentive et leurs remarques pertinentes et constructives.

Un grand merci également à Nicolas Chopin, toujours là pour répondre patiemment à mes questions. Son enthousiasme m'a convaincu de faire une thèse quand j'étais son élève à l'ENSAE.

Merci aux chercheurs rencontrés en cours de route, notamment Arnaud Doucet qui m'a accueilli trois mois à Vancouver et grâce à qui ma motivation a triplé. Merci aussi à lui pour avoir partagé avec moi sa vision du monde académique, et pour les tournées de Stella à Yaletown.

Merci à Pierre Del Moral pour sa disponibilité, son soutien, et son envie de partager.

Merci en général à tous les chercheurs bienveillants envers les étudiants, comme Eddy Campbell, Gersende Fort, Ajay Jasra, Jean-Michel Marin, Eric Moulines, Eric Parent, Gareth Peters et Nick Whiteley. J'espère travailler avec eux à l'avenir.

Merci à Luke Bornn pour les bons moments passés à Vancouver, dans l'Utah et à Bordeaux, et les échanges optimistes sur l'avenir de la statistique. Merci à Julien Cornebise pour les milliers de conseils geek et les avis passionnés sur les donuts canadiens.

Merci à Anthony Lee et Lawrence Murray pour deux bons mois à Perth, plein d'enseignements, de planctons et de quakkos.

Merci à Céline Duval, Jean-Bernard Salomond et Christian Schäfer pour avoir mis l'ambiance dans le bureau F14, ainsi qu'aux autres jeunes chercheurs du CREST, toujours motivés pour un "vendredibière" ou un "lundirouge". Merci à Robin Ryder pour l'ambiance thé et gâteaux dans le bureau F15, et à Philippe Dupuis pour les pauses mots-croisés. Merci aussi aux chercheurs permanents du CREST pour leurs conseils et les discussions intéressantes pendant les pauses-déjeuner: Pierre Alquier, Patrice Bertail, Eric Gautier, Marc Hoffmann et Judith Rousseau.

Merci enfin à mes proches pour leur soutien.

Résumé

Cette thèse présente différentes contributions aux méthodes de Monte Carlo utilisées en statistique bayésienne. Le paradigme bayésien constitue l'une des principales approches actuelles de la statistique et dispose désormais d'une méthodologie importante pour l'inférence et le choix de modèle. Néanmoins, au fur et à mesure que les modèles statistiques deviennent plus réalistes et qu'ils s'écartent des hypothèses classiques de normalité et de linéarité, le calcul de diverses quantités nécessaires à l'analyse statistique devient un problème en soi. En particulier, des intégrales en grande dimension, dans lesquelles les intégrandes peuvent être fortement multimodales, doivent être approchées de manière efficace. Par ailleurs, l'évaluation de l'intégrande en chaque point de l'espace peut nécessiter un calcul conséquent, ce qui résulte en des procédures d'intégration très coûteuses. Ces intégrales sont en général approchées au moyen des méthodes de Monte Carlo, qui requièrent une aptitude générale à simuler des échantillons selon des lois de probabilité. Le premier chapitre de ce document explique ce contexte et passe en revue les techniques de Monte Carlo les plus génériques. Les chapitres suivants visent à améliorer certaines de ces techniques, à en proposer de nouvelles et à étudier leurs propriétés théoriques, dans le contexte de l'échantillonnage de distributions multimodales dont les densités sont parfois coûteuses à évaluer.

Un résumé des différents chapitres est proposé ci-dessous; une introduction en français est également présentée au début de chaque chapitre.

Chapitre 1 : l'introduction commence par illustrer l'omniprésence d'intégrales de grande dimension en statistique bayésienne. La notion de loi cible est introduite, puis certaines des méthodes de Monte Carlo les plus génériques sont présentées, en insistant sur les pré-requis de chaque méthode en termes de connaissance sur la loi cible. Enfin les principales difficultés dont cette thèse traite dans les chapitres suivants sont présentées: l'éventuelle multimodalité de la loi cible et le coût élevé associé à l'évaluation de sa densité de probabilité.

Chapitre 2 (co-écrit avec Christian P. Robert & Murray Smith): une méthode est proposée pour réduire la variance des estimateurs fondés sur l'échantillon généré par l'algorithme de Metropolis–Hastings à proposition indépendante. Inspirée des techniques dites de Rao–Blackwellisation, la méthode réduit la variance en intégrant partiellement deux sources de variabilité: l'ordre dans lequel les propositions sont fournies à l'algorithme, et les variables aléatoires uniformes qui sont tirées pour accepter ou rejeter les propositions. La méthode a été pensée pour tirer parti de processeurs parallèles, contrairement à d'autres méthodes existantes de réduction de variance. La méthode est illustrée sur un exemple simple de régression "probit" où des gains significatifs sont obtenus, notamment lorsque le taux d'acceptation de l'algorithme est faible.

Chapitre 3 (co-écrit avec Nicolas Chopin): une méthode de Monte Carlo séquentielle est proposée, telle qu'au cours de l'algorithme, les régions de l'espace qui ont déjà été visitées par les particules sont pénalisées, alors que les régions qui n'ont pas été visitées sont favorisées. Les régions sont définies en partitionnant l'espace selon un

axe appelé “coordonnée de réaction”. La méthode a pour but de simuler efficacement des échantillons selon des lois présentant une forte multimodalité dans leur densité de probabilité. La méthode peut être vue comme l'équivalent particulaire de certains algorithmes de Monte Carlo à chaîne de Markov conçus pour répondre au problème de la multimodalité dans la loi cible. L'algorithme est illusté numériquement à travers deux modèles de mélange.

Chapitre 4 (co-écrit avec Robin J. Ryder): une analyse de l'algorithme de Wang–Landau est proposée. Cet algorithme, de type Monte Carlo à chaîne de Markov adaptatif, est précisément conçu pour générér un échantillon suivant une loi fortement multimodale. Le chapitre propose l'étude d'une variante de l'algorithme utilisant le critère stochastique dit de “Flat Histogram”, et établit la validité théorique de cette variante en montrant que le critère est toujours atteint en un temps d'espérance finie. La preuve est fondée sur des hypothèses fortes sur l'espace d'état et sur la loi cible, qui sont ensuite discutées. Enfin les conséquences du résultat sont illustrées sur un exemple jouet.

Chapitre 5 (co-écrit avec Luke Bornn, Arnaud Doucet & Pierre Del Moral): diverses améliorations méthodologiques sont apportées à l'algorithme de Wang–Landau, notamment l'utilisation de plusieurs chaînes au lieu d'une seule, et l'adaptation de certains paramètres algorithmiques. Une stratégie est ainsi proposée pour modifier la partition de l'espace d'état au cours de l'algorithme, ce qui résout en partie la difficulté de choisir cette partition. Ces modifications mènent à un algorithme générique pour explorer les lois multimodales, qui se montre efficace sur une série d'exemples d'inférence bayésienne et sans que de fastidieux réglages préalables soient nécessaires.

Chapitre 6 (co-écrit avec Nicolas Chopin & Omiros Papaspiliopoulos): une variante séquentielle des méthodes de Monte Carlo à chaîne de Markov particulaires (pMCMC) est proposée. Elle permet de simuler séquentiellement selon la loi a posteriori des paramètres d'un modèle à chaîne de Markov cachée, ainsi que selon la loi de la chaîne cachée. Le caractère séquentiel permet d'obtenir une nouvelle approximation particulaire de la loi a posteriori à l'apparition de chaque nouvelle observation. Par ailleurs la méthode permet d'obtenir des approximations des lois de filtrage et de lissage tenant compte de l'incertitude sur les paramètres, ainsi qu'un estimateur de l'évidence du modèle, utile au calcul des facteurs de Bayes. L'algorithme est illustré sur deux exemples: un modèle de volatilité stochastique appliqué aux données S&P 500, et un modèle pour lequel l'équation de mesure est donnée par une loi généralisée de valeurs extrêmes, appliqué à des données de records athlétiques.

Chapitre 7 : la conclusion de cette thèse présente des perspectives de recherche, en vue d'une meilleure compréhension des algorithmes de Monte Carlo à chaîne de Markov adaptatifs utilisant plusieurs chaînes, tels que celui présenté dans le chapitre 5, ainsi que dans la direction de l'inférence bayésienne dans les modèles à chaîne de Markov cachée, pour lequels le cadre introduit par le chapitre 6 semble prometteur.

Summary

Computational contributions to Bayesian statistics

This thesis presents contributions to the Monte Carlo methodology used in Bayesian statistics. The Bayesian framework is one of the main approaches to statistics and includes a rich methodology to perform inference and model choice. However, as statistical models become more realistic and drift away from the classical assumptions of normality and linearity, computing some of the quantities involved in the statistical analysis becomes a challenge in itself. In particular high-dimensional integrals have to be efficiently approximated, where the integrands can be highly multimodal. Moreover each point-wise evaluation of the integrands can require a lot of computational effort, which results in expensive integration schemes. These integrals are typically approximated using Monte Carlo methods, requiring the ability to sample from general probability distributions. The first chapter of this document explains this motivating context and reviews some of the most generic Monte Carlo techniques. The following chapters aim at improving some of these techniques, at proposing new methods and at analysing their theoretical properties, in the context of sampling from multimodal and computationally expensive probability distributions.

A summary of each chapter is provided below.

Chapter 1 first presents some motivation by illustrating the omnipresence of very high-dimensional integrals in Bayesian statistics. The notion of target distribution is introduced; then an overview of generic Monte Carlo methods is presented, where emphasis is laid on the specific knowledge of the target distribution required by each method. Finally we give a description of the specific issues that this thesis deals with: multimodality and expensive target density evaluation.

Chapter 2 (joint work with Christian P. Robert & Murray Smith) presents a method to decrease the variance of estimates based on the output of the Independent Metropolis–Hastings algorithm. Using Rao–Blackwellization, the method decreases the variance by partly integrating out two sources of variability: the order in which the proposals are processed by the algorithm, and the uniform random variables drawn to accept or reject these proposals. The method is intended to be easy to implement on parallel computing devices, as opposed to some previous variance reduction techniques for Markov chain Monte Carlo methods. The variance reduction is illustrated on a simple probit regression example and substantial gains are observed, especially when the acceptance rate of the initial algorithm is low.

Chapter 3 (joint work with Nicolas Chopin) presents a modification of the Sequential Monte Carlo (SMC) sampler, such that regions of the state space that have already been visited by the particles are penalized during the run, in order to favour the exploration of the state space. This method aims at improving the performance of sampling methods when the target density function is highly multimodal. It can be seen as the particle counterpart of some Markov chain Monte Carlo algorithms designed to address the same multimodality issue. The proposed algorithm is illustrated on two mixture model examples.

Chapter 4 (joint work with Robin J. Ryder) considers the Wang–Landau algorithm, one of the Markov chain Monte Carlo algorithms motivated by the issue of multimodality in the target distribution. The chapter proposes a proof that a variation of the algorithm, using the so-called Flat Histogram criterion, is valid in the sense that this stochastic criterion is met in finite time. The proof relies on strong assumptions on the state space and the target distribution, which are discussed. The result is illustrated on a toy example.

Chapter 5 (joint work with Luke Bornn, Arnaud Doucet & Pierre Del Moral) considers algorithmic improvements of the Wang–Landau algorithm, using multiple chains and adapting the tuning parameters using the generated sample. A strategy is proposed to modify the partition of the state space in an adaptive manner, thus partly alleviating the practical difficulty of designing such a partition, which is an essential tuning parameter of the Wang–Landau algorithm. The proposed improvements lead to a general-purpose algorithm to explore multimodal target density functions, which proves efficient on a range of Bayesian inference problems without requiring tedious algorithmic tuning from the user.

Chapter 6 (joint work with Nicolas Chopin & Omiros Papaspiliopoulos) presents a sequential Monte Carlo method to sample from the posterior distribution of general Hidden Markov models. Similarly to particle Markov chain Monte Carlo methods, the method allows to sample from the posterior distribution of both the parameters and the latent process; it does so in a sequential manner, providing particle approximations at each time step. As an aside, it can be used to approximate the filtering and smoothing distributions under parameter uncertainty, and it provides an estimate of the model evidence. The method is illustrated on two examples: a stochastic volatility model on the S&P 500 dataset and a model with measurements following a generalized extreme value distribution, applied to a dataset of athletic records.

Chapter 7 presents some future lines of research, towards a better understanding of population-based adaptive Markov chain Monte Carlo algorithms such as the one introduced in Chapter 5, and more efficient tools to perform Bayesian inference and model choice in hidden Markov models, based on the sequential framework introduced in Chapter 6.

Contents

Chapter 1

Introduction

> *Deciding which information to extract, and sifting through massive amounts of information to find what is useful, was something only a flesh-and-blood person could do.*
>
> – Haruki Murakami, *1Q84*

Diverses situations motivées par l'approche bayésienne mènent à de difficiles calculs d'intégration. En particulier, en absence de conjugaison entre la loi *a priori* et la vraisemblance du modèle statistique, l'inférence nécessite d'approcher numériquement l'intégrale de différentes fonctions par rapport à la loi *a posteriori*. Par ailleurs, certains modèles statistiques mènent naturellement à l'étude d'intégrales de grande dimension. Les modèles à chaîne de Markov cachée en constituent un exemple particulièrement utile à l'analyse des séries temporelles. Dans ces modèles, les objets d'intérêt sont communément exprimés en termes de lois de probabilité (loi de filtrage, loi de lissage), et l'inférence requiert l'intégration par rapport à ces lois. Naturellement, la combinaison de l'approche bayésienne et de ces modèles mène à des calculs particulièrement difficiles, puisque deux niveaux d'intégration sont à prendre en compte: l'intégration sur l'espace des paramètres et l'intégration sur l'espace des états latents.

Dans ce contexte, les méthodes de Monte Carlo pour l'approximation d'intégrales apportent une aide précieuse à l'inférence statistique. Néanmoins la mise en pratique de ces méthodes dépend de la capacité de l'utilisateur à obtenir des échantillons suivant une quelconque loi de probabilité d'intérêt. Malheureusement, différentes raisons peuvent rendre cette simulation difficile; notamment la densité de probabilité peut être impossible ou très chère à calculer exactement, ou elle peut présenter une forte multimodalité, ce qui motive la recherche de nouvelles méthodes de simulation et l'amélioration des techniques existantes. Ces méthodes de Monte Carlo peuvent être classées en fonction des pré-requis à leur mise en œuvre: certaines méthodes nécessitent de pouvoir évaluer la densité de la loi en tout point, à une constante multiplicative près; d'autres requièrent uniquement la possibilité d'en obtenir un estimateur sans biais; d'autres enfin ne nécessitent que la possibilité de simuler des données synthétiques suivant le modèle statistique. Dans ce chapitre certaines méthodes de Monte Carlo sont décrites en insistant sur leurs pré-requis, afin que les méthodes proposées dans les chapitres suivants puissent être placées dans le contexte général des méthodes de simulation.

Enfin, les problèmes de la multimodalité et du coût associé à l'évaluation de la densité cible sont discutés puisqu'ils forment le fil conducteur des chapitres suivants. Le problème du coût computationnel dans l'évaluation de la loi cible est présenté avec les approches qui ont été proposées pour y répondre, notamment les techniques de Rao–Blackwellisation et les dernières avancées en calcul parallèle. La multimodalité est illustrée à travers deux exemples: un modèle de mélange gaussien pour lequel la densité cible est calculable en tout point, et un modèle de dynamique des populations pour lequel seulement un estimateur sans biais de la densité cible est disponible.

The introduction describes the general framework, common to the following chapters: it starts by introducing the notion of target distribution, then proceeds to a quick overview of Monte Carlo techniques and finally underlines some specific issues that the following chapters address in more details.

1.1 Target distributions

The expression "target distribution" has no precise mathematical definition; it usually refers to a probability distribution of special interest, for it contains all the relevant information regarding the problem at hand. In this section Bayesian statistics and Hidden Markov models are introduced to provide examples of situations where the objects of interest are probability distributions, and will hence be referred to as target distributions.

1.1.1 Bayesian posterior distributions

One of the main paradigms in the current statistical literature, Bayesian statistics is particularly filled with references to the phrase "target distribution". An extensive introduction to Bayesian statistics and its benefits is provided by [Robert 94]. Bayesian statistics is a popular approach to virtually every aspect of statistics. It allows to translate:

- a data set,
- a statistical model with unknown parameters,
- a probability distribution on the parameters, prior to taking the data into account (the *prior distribution*),

into a probability distribution combining these three elements, referred to as the *posterior distribution*. This distribution then acts as a building block for each step of the statistical analysis: parameter inference, prediction, hypothesis testing and so forth. Note that if we allow the parameters to be infinite dimensional, the above description encompasses parametric and non-parametric Bayesian statistics.

For simplicity let us consider a simple case of parametric model, that is, the parameter is d-dimensional with $d \in \mathbb{N}$. The data set is a collection of N values $(y_1, \ldots, y_N) \in \mathcal{Y}^N$ where $\mathcal{Y}$ is called the sample space. These are supposed to be realisations of random variables $(Y_1, \ldots, Y_N)$ defined on the same space: it is hence assumed that there are a σ-algebra Σ_Y and a measure λ such that $(\mathcal{Y}, \Sigma_Y, \lambda)$ is a measure space. In this case λ is the reference measure, hereafter simply denoted by dy. The statistical model defines the law of the random variables $(Y_n)_{n=1}^N$, that is typically called the *likelihood*, denoted by $\mathcal{L}$. Here we assume that this law admits a density with respect to the reference measure dy. The likelihood can be seen as a function of the possible realisations $(y_1, \ldots, y_N)$ and as a function of parameters of the statistical model, which we denote by $\theta \in \Theta$, where typically $\Theta \subset \mathbb{R}^d$:

$$\mathcal{L} : \theta \times (y_1, \ldots, y_N) \mapsto \mathcal{L}(\theta; y_1, \ldots, y_N)$$

In the following we assume that the realisations $(y_1, \ldots, y_N)$ are given. Up to here the setting is common to most of the statistical literature.

The specificity of the Bayesian approach is to consider the parameter θ to be a random variable as well. It requires Θ to be associated with a measure space, say $(\Theta, \Sigma_\Theta, d\theta)$ (with explicit notation). We put a distribution on θ, called the *prior* distribution, with the prior probability density function with respect to $d\theta$ denoted by p. The object of interest is the posterior distribution, that is the conditional distribution of the parameters given the observations, whose density is denoted by π and is obtained through Bayes formula as follows:

$$\pi(\theta|y_1,\dots,y_N) = \frac{p(\theta)\mathcal{L}(\theta;y_1,\dots,y_N)}{\int_\Theta p(\theta)\mathcal{L}(\theta;y_1,\dots,y_N)d\theta} \tag{1.1}$$

Provided the denominator is finite, the posterior distribution is well-defined. From this distribution, many by-products of interest can be constructed, depending on the application. Parameter estimation is achieved for instance through

$$\mathbb{E}\left(\theta|y_1,\dots,y_N\right) = \int_\Theta \theta\ \pi(\theta|y_1,\dots,y_N)d\theta$$

which is the posterior expectation, or through

$$\text{argmax}_{\theta\in\Theta}\pi(\theta|y_1,\dots,y_N)$$

which is the *maximum a posterior*. Tests on the parameter values involve computing terms of the form:

$$\int_{\Theta_i} \pi(\theta|y_1,\dots,y_N)d\theta$$

for $i = 0, 1$, where sets Θ_0, Θ_1 are subsets of Θ corresponding to the null hypothesis and the alternative hypothesis respectively. Prediction under parameter uncertainty can involve the following distribution:

$$p(y|y_1,\dots,y_N) = \int_\Theta f(y|\theta,y_1,\dots,y_N)\pi(\theta|y_1,\dots,y_N)d\theta$$

where f here denotes the density of a new observation (the *predictive density*), given previous observations and a parameter. Out of this distribution, one might want to compute the expectation to use it as a predictor for a future observation, along with a credible interval. Integrating the parameters with respect to the posterior distribution allows to make a fully Bayesian prediction, without having to estimate the parameters first and then to plug the estimated value into the predictive density. As in the literature we say that this approach *takes parameter uncertainty into account.*

Model choice can involve the following quantity which is usually called the evidence:

$$\mathcal{E} = \int_\Theta p(\theta)\mathcal{L}(\theta;y_1,\dots,y_N)d\theta$$

and which is the denominator in Equation 1.1. Again, since this quantity integrates over the parameter space, it provides an interesting quantity to choose between models without using two-step methods, where first a most plausible parameter value is estimated for each model and then models fitted on these parameter values are compared one to another.

This simple parametric setting illustrates that under the Bayesian paradigm, many quantities can be expressed as functionals of the posterior distribution (integrals in many cases). Therefore the posterior distribution will be seen as a “target distribution” when considering the numerical methods of the following chapters, and the ability to sample from it will allow to approximate the various quantities described above, as will be explained in Section 1.2. Of course depending on the statistical model and the data, the difficulty of approximating these quantities by sampling from the posterior distribution is very uneven, and closer attention will be paid to some challenging cases in Section 1.3.

1.1.2 Hidden Markov Models

An interesting class of statistical models is referred to as Hidden Markov Models (HMM), which can be studied under the Bayesian framework but not necessarily. HMMs constitute a flexible class to model time series, that is, a sequence of ordered, dependent random variables. More specifically, we are going to consider models where the observed time series $(Y_t)_{t\geq 0}$ is assumed to be a noisy measurement of an underlying, unobserved Markov process denoted by $(X_t)_{t\geq 0}$ (hence the name "Hidden Markov model"). In general their mathematical formulation goes as follows: first introduce a set $\mathcal{X}$ on which the hidden process lives and a set $\mathcal{Y}$ on which the observations live; both spaces are supposed to be associated with σ-algebras (Σ_X and Σ_Y respectively) and reference measures (dx and dy respectively). Suppose that X_0 is a random variable in $\mathcal{X}$ following a distribution admitting a density μ_0 with respect to dx. Then, for any $t \geq 1$ we define X_t as a random variable whose distribution might only depend on the realisation of X_{t-1}, denoted by x_{t-1}. The sequence $(X_t)_{t\geq 0}$ thus constitutes a Markov chain. We denote by f_θ the density of the law of X_t given X_{t-1}, with respect to dx. This law is often called the *transition*. Then we assume that the law of the observation Y_t depends only on the value of X_t, through the *measurement* distribution, admitting a density with respect to dy denoted by g_θ. Note that the transition and measure probability density functions are indexed by a parameter $\theta \in \Theta \subset \mathbb{R}^d$, which is considered as known or unknown depending on the situation.

In these models, various questions naturally arise: first, can we estimate the parameters given the observations ? Writing the likelihood for this class of models shows the difficulty of this task.

$$\mathcal{L}(\theta; y_1, \ldots, y_T) = \int_{\mathcal{X}^T} \mu_0(x_0) \prod_{t=1}^{T} f_\theta(x_t|x_{t-1}) g_\theta(y_t|x_t) dx_{0:T} \tag{1.2}$$

using the short notation $x_{l:k} = (x_l, \ldots, x_k)$ which is standard in the HMM literature. This integral is on a $(T+1) \times \dim(\mathcal{X})$ dimensional space, and hence its dimension increases with the number of observations T. Without restrictive assumptions on f_θ and g_θ, there is no analytical form of this integral, and hence the likelihood is not tractable. Thus finding its maximum or approximating the posterior distribution in a Bayesian framework are considered challenging problems (and increasingly challenging as T increases). In the Bayesian framework, a recent breakthrough occurred in 2010 ([Andrieu 10]), allowing to sample from the posterior distribution provided that f_θ can be sampled from and that g_θ can be evaluated up to a normalizing constant.

Secondly, other tasks than parameter inference can be of interest, most notable examples being filtering, prediction and smoothing (see Section 3.1 of [Cappé 05]). In particular filtering means estimating the distribution of X_t given the observations up to time t, smoothing is the problem of estimating the distribution of X_k given the observations up to time t, for $k < t$. Early references to these problems mostly considered the linear Gaussian case, where the transition and measurement densities are Gaussian. In this case the three aforementioned problems are solved by the Kalman filter ([Kalman 61]), that provides the exact distributions of interest, which are all Gaussian. In full generality, these distributions do not admit such closed-form representation. However they admit useful recursion formulae, leading to efficient approximation schemes. For instance, if we consider the filtering problem (for a fixed parameter value θ), the following calculation links the filtering distribution at time t with the filtering distribution at time $t+1$:

$$\begin{aligned} p_\theta(x_{t+1}|y_{1:t+1}) &= p_\theta(x_{t+1}|y_{1:t}, y_{t+1}) \\ &= \frac{p_\theta(y_{t+1}|x_{t+1}, y_{1:t})p_\theta(x_{t+1}|y_{1:t})}{p_\theta(y_{t+1}|y_{1:t})} \\ &= \frac{g_\theta(y_{t+1}|x_{t+1}) \int_{\mathcal{X}} f_\theta(x_{t+1}|x_t)p_\theta(x_t|y_{1:t})dx_t}{p_\theta(y_{t+1}|y_{1:t})} \end{aligned} \tag{1.3}$$

Except for the transition and measure densities denoted by f_θ and g_θ, the other densities above are denoted by p_θ but their arguments should make their differences explicit: for instance $p_\theta(x_{t+1}|y_{1:t})$ denotes the density of X_{t+1} given $Y_{1:t} = y_{1:t}$ and a given parameter value θ, evaluated at $x_{t+1} \in \mathcal{X}$. This formula makes it apparent that there are as many $\dim(\mathcal{X})$-dimensional integrals to compute as there are time steps T, since each filtering recursion involves an integral. Nonetheless it allows to break the initial large-dimensional integral into a sequence of one-dimensional ones. Errors occurring in the approximation of each integral might (or not) propagate through the recursion, making the filtering problem challenging in general. Note also that the denominator $p_\theta(y_{t+1}|y_{1:t})$ is an interesting quantity in itself, referred to as the incremental likelihood at time $t+1$, but it is often unavailable in closed-form. The product over time of these incremental likelihoods leads to the likelihood defined in Equation 1.2. Sequential Monte Carlo methods are considered efficient and generic methods to approximate the filtering distributions and also provide unbiased estimates of the likelihood, for a given parameter value (see [Doucet 01]).

Going back to the Bayesian framework, the filtering distribution *under parameter uncertainty* is perhaps an even more useful distribution to look at, since it takes into account the lack of perfect knowledge about the parameter values. This distribution is linked with the former filtering distribution as follows:

$$\begin{aligned} p(x_t|y_{1:t}) &= \int_\Theta p(x_t, \theta|y_{1:t})d\theta \\ &= \int_\Theta p(x_t|y_{1:t}, \theta)p(\theta|y_{1:t})d\theta \\ &= \int_\Theta p_\theta(x_t|y_{1:t})\pi(\theta|y_{1:t})d\theta \end{aligned}$$

Since the filtering distribution *under parameter uncertainty* is an integral of the former filtering distribution with respect to the posterior distribution of the parameters, approximating it is even more challenging. [Andrieu 10] and Chapter 6 introduce methods to sample from the joint distribution of the hidden process $(X_t)_{t\geq 0}$ and the parameter θ given the observations, hence allowing to approximate the filtering distribution *under parameter uncertainty.*

To summarize, for HMMs and even when the parameter is considered as known, the natural objects of interest are distributions, and hence we will refer to them as target distributions, similarly to the posterior distribution in Bayesian statistics.

1.2 Monte Carlo methods

Having seen that distributions are objects of interest in various contexts, we will argue in this section that getting samples from those distributions is an efficient mean to compute integrals with respect to them. In Section 1.2.1 we will recall the basic Monte Carlo method to compute integrals, which requires the ability to sample from a generic probability distribution. We will then see in Section 1.2.2 a brief list of methods to sample from a generic distribution, and in Section 1.2.3 we will argue for Monte Carlo methods in general by comparing them with alternative methods to approximate integrals through a few toy examples.

1.2.1 Basic Monte Carlo

Let us first recall the most basic Monte Carlo method. Consider the problem of computing an integral $\mathcal{I}(\varphi)$ defined as:

$$\mathcal{I}(\varphi) = \int_{\mathcal{X}} \varphi(x)dx$$

where φ is some function on a space $\mathcal{X}$ such that the integral is finite. There are a function h and a probability density function f such that

$$\mathcal{I}(\varphi) = \int_{\mathcal{X}} h(x)f(x)dx = \mathbb{E}_f\left[h(X)\right]$$

The basic Monte Carlo method assumes that it is possible to obtain a collection of N independent draws from f, and that one can compute h point-wise. We denote the draws from f by $X_1, \ldots, X_N$. Then the Monte Carlo estimator of $\mathcal{I}(\varphi)$ is defined as:

$$\widehat{\mathcal{I}(\varphi)}^N := \frac{1}{N}\sum_{n=1}^{N} h(X_n)$$

The Monte Carlo estimator is unbiased, strongly consistent by the Law of Large Numbers (LLN), and satisfies the following Central Limit Theorem (CLT):

$$\sqrt{N}\left(\widehat{\mathcal{I}(\varphi)}^N - \mathcal{I}(\varphi)\right) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}\left(0, \mathbb{V}_f\left[h(X)\right]\right)$$

if $\mathbb{V}_f\left[h(X)\right]$ is finite.

In general, the representation of a generic function φ as a product of a function h and a probability density function f is only artificial. In the statistics framework however, this representation is very natural (see the examples of integrals of interest in Section 1.1): the distribution f corresponds to a target distribution, and is hence given by the problem; the choice of h leads to the computation of various quantities describing the target, typically its moments.

From this basic Monte Carlo method, a lot of improvements and generalizations have been proposed (see *e.g.* [Robert 04]). We will be mostly concerned with the ability to sample from the distribution f. Accordingly Section 1.2.2 considers situations where obtaining independent and identically distributed (iid) samples from the target distribution is challenging.

1.2.2 Sampling from any distribution

Sampling from a distribution means getting a set of realisations (a sample) of random variables following that distribution. Therefore a sampling algorithm presents a way to explicitly construct random variables such that in practice a set of realisations can be generated. For well-known, ubiquitous distributions (*e.g.* exponential, Gaussian, gamma, etc) many techniques have been found to get iid samples, see *e.g.* [Devroye 85] for an exhaustive list. These methods rely on the ability to sample (iid) from the uniform distribution on the interval $[0, 1]$. This ability will be taken for granted in the rest of this thesis, and whenever uniform draws are needed a standard pseudo-random number generator (RNG) is used. RNGs go back to [Von Neumann 51] and still constitute an active research area, see for instance the recent parallel-friendly algorithms of [L'Ecuyer 01] and [Marsaglia 03]. Other recent techniques to obtain samples mimicking uniform draws include Herding, introduced by [Welling 09] and already extended multiple times in *e.g.* [Chen 10, Bach 12].

For a generic probability distribution, there might not be obvious ways to obtain an iid sample. When classical methods (*e.g.* inverse transform sampling or rejection sampling) do not apply, one can still resort to a variety of methods, some of which (the most generic to our knowledge) are described in the following sections. We will not describe the methods themselves, but instead we will emphasize the requirements of each method, in terms of degrees of knowledge about the distribution. For algorithmic details, Chapter 2 starts with a description of the Metropolis–Hastings algorithm and Chapter 3 with Sequential Monte Carlo (SMC) samplers. Chapter 6 provides descriptions for both SMC samplers and SMC for filtering, as well as for particle MCMC.

We note that with the exception of Perfect Sampling algorithms [Propp 96, Casella 01], the latest Monte Carlo methods hence produce dependent and approximate samples, seeking only guarantees that the error in our approximation based on the generated sample goes to zero when the sample size grows. The methods described below actually either produce Markov chains or collections of dependent and weighted random variables.

Tractable probability density function

In this first case, we assume that the target distribution admits a probability density function (pdf) that can be evaluated point-wise, at least up to a multiplicative constant; we then say that the pdf is tractable. This is a common case in the Markov Chain Monte Carlo literature (MCMC, see *e.g.* [Robert 04] as well as [Gelman 10] for recent developments). In Bayesian statistics, provided that the prior pdf and the likelihood can be evaluated point-wise, then the posterior pdf obviously falls into that category, where the unknown multiplicative constant is the normalizing constant of the posterior distribution, *i.e.* the denominator in Equation 1.1.

The Metropolis–Hastings algorithm (MH, see [Metropolis 53] and [Hastings 70] for the seminal papers) is one of the most generic MCMC algorithm applicable in this situation. Given a target pdf denoted by f and some tuning parameters, it produces a Markov chain $(X_t)_{t\geq 1}$ (associated with an initial distribution μ, and a transition kernel P) whose marginal distribution converges to the target distribution. Similarly to LLNs and CLTs for iid samples, ergodic theorems guarantee the validity of the method. For instance, under conditions on the target pdf and the tuning parameters (see [Tierney 94], Corollaries 3 and 4 for example), the generated chain can be proven to be geometrically ergodic, *i.e.* there exists a positive function M integrable with respect to f and a positive constant $r < 1$ such that:

$$\|P^n(x,\cdot) - f(\cdot)\| \leq M(x)r^n$$

where $\|\cdot\|$ denotes the total variation norm, for any starting point x and any integer n, where P^n is the n-th iterate of the kernel P. Under stronger conditions, we can suppress the dependence on the starting point x. Most importantly, ergodicity implies stability of time-averages. For a function h such that $|h|$ is integrable with respect to the target pdf f, ergodicity implies:

$$\frac{1}{T}\sum_{t=1}^{T} h(X_t) \xrightarrow[T\to\infty]{a.s.} \mathbb{E}_f(h(X))$$

and the CLT for Markov chains:

$$\sqrt{T}\left(\frac{1}{T}\sum_{t=1}^{T}h(X_t)-\mathbb{E}_f(h(X))\right)\xrightarrow[T\to\infty]{Law}\mathcal{N}(0,\sigma^2(h))$$

for a positive constant $\sigma^2(h)$ independent of T. Quite notably, these results do not depend on the initial law μ of the Markov chain, and hence do not assume that the Markov chain is stationary, as noted in section 1.8 of [Gelman 10]. Most of the results on Markov chain that are useful for MCMC are provided in the exhaustive book [Meyn 93], while a useful summary for MCMC practitioners is provided in [Nummelin 02].

The obtained sample is not iid: indeed it is both dependent (as any non-trivial Markov chain) and not necessarily distributed according to the target distribution, as this might only be the case asymptotically. However theoretical results like the CLT for Markov chains provide a solid ground to use the generated Markov chain exactly as an iid sample, in order to approximate integrals of interest. Accordingly in the MCMC literature, the term "sample" does not necessarily refer to an iid sample, but encompasses Markov chains as well. Therefore the main limitation of the method is not in the dependence between the generated points, but in the potentially expensive computational cost, as well as in the reported poor performance when the target pdf is highly multimodal. These issues are illustrated in Section 1.3.

Estimable probability density function without bias

A recent breakthrough (originally proposed in [Beaumont 03] and studied thoroughly as a generic family of MCMC algorithm in [Andrieu 09]) allows to extend the applicability of MCMC to cases where the target pdf is not available but can be unbiasedly estimated pointwise, up to a multiplicative constant. These methods are referred to as *pseudo-marginal* methods.

Let us introduce a toy example as in [Andrieu 09] to illustrate those new methods. Suppose that the target pdf can be written as an integral, as follows:

$$\forall x\in\mathcal{X}\quad f(x)=\int_{\mathcal{Y}}p(x,y)dy$$

for some joint pdf p defined on $\mathcal{X}\times\mathcal{Y}$. If the integral is not tractable for any x, then we cannot compute $f(x)$ for any x, even up to a multiplicative constant, and hence we cannot use a MH algorithm to sample from f. We will refer to this impractical, but worth seeking MH as the ideal MH algorithm. Now suppose that we can get a sample $(y_1,\ldots,y_N)$ from a distribution admitting a density q_Y, and that we can evaluate the joint density p. Then we have access to the following estimate:

$$\hat{f}^N(x)=\frac{1}{N}\sum_{n=1}^{N}\frac{p(x,y_n)}{q_Y(y_n)}$$

which is unbiased:

$$\mathbb{E}_{q_Y}\left[\hat{f}^N(x)\right]=f(x)$$

In order to mimick the ideal MH algorithm, a workaround to the intractability of f consists in simply replacing each evaluation $f(x)$ by the estimate $\hat{f}^N(x)$, leading to an "approximate" MH. If the estimates are precise enough, intuitively the approximate algorithm will behave like its ideal counterpart; however an approximation is introduced, and as a consequence, the generated sample might not be a Markov chain admitting the target f as an invariant distribution. This intuition is generally true but it turns out that some variations of these approximate algorithms (for instance Beaumont's Grouped Independence MH (GIMH) introduced in [Beaumont 03]), while replacing the true target pdf evaluations by estimates, still produce a Markov chain admitting the exact target distribution f as an invariant distribution, just like the ideal algorithm. In other words, it is possible to produce such Markov chains without being able to evaluate the target pdf at all.

Of course, the quality of the approximation of $f(x)$ by $\hat{f}^N(x)$ does matter, and this quality is here parameterized by the integer N counting the number of draws from q_Y used in each estimation of $f(x)$. [Andrieu 09] notes (in the example section) that using too small a value for N leads to very poor results (*e.g.* very low acceptance rate), while as soon as N is bigger than some value, the quality of the results does not seem to depend heavily on the exact value of N. In other words for a given target density function, there seems to be a minimal value N_0 such that, provided $N > N_0$, the approximate algorithm behaves very much like its ideal counterpart; even though for any N, theoretical results guarantee the convergence of time-averages. Some fascinating links between the performance of the ideal algorithm and the performance of the approximate version are obtained in [Andrieu 09].

The particular case of HMMs leads to the ability to sample from the posterior distribution of the parameters for a large class of models, using particle MCMC methods as introduced in [Andrieu 10]. Chapter 6 fits in this exact framework as well. Regarding the role of N in the performance of approximate versions of MH algorithms, [Andrieu 10] provides guidelines on how to choose N as a function of the number of observations, and in Chapter 6 we justify an automatic method to tune N along the run.

These new methods, and particularly the particle MCMC of [Andrieu 10], already prove to be useful in a variety of settings. Indeed they extend the applicability of the MH algorithm, thus allowing to use an already well-understood framework in cases where only custom-made sampling methods, if any, were applicable. These settings include ecology [Peters 10], biochemical network modelling [Golightly 11], epidemiology [Rasmussen 11], finance [Peters 11, Bauwens 11]. This motivates further methodological improvements, as well as better understanding of the relative properties of approximate versions compared to ideal algorithms. Conversely, since these new particle MCMC methods rely on the MCMC methods, they eventually suffer from the same limitations, for instance in case of multimodality in the target distribution as will be seen in Section 1.3.

Estimable log probability density function

Let us mention a promising case described by Nicholls [Nicholls 12]: here the target distribution is such that there are normally distributed estimates of the target log density values, at any point. It hence differs from the case described in the previous section where the density itself was estimated. This case occurs in physics where the energy, corresponding to minus the log density, is noisy; this is the motivation of [Ceperley 99] who introduce a Metropolis–Hastings suited to this case. Of course, requiring that the estimate has to be normally distributed is restrictive in practice, where the estimate would typically be asymptotically normally distributed, in the best case. Indeed suppose that the observations $(Y_1, \dots, Y_N)$ are iid from the density $f(\cdot|\theta)$, then the log likelihood can be written as:

$$\ell(\theta; y_1, \dots, y_N) = \log \mathcal{L}(\theta; y_1, \dots, y_N) = \sum_{n=1}^{N} \log f(y_n|\theta)$$

then all the terms $\log f(y_n|\theta)$ are independent, and hence the estimate $\widehat{\ell(\theta)}_m$ defined for $m \leq N$ as:

$$\widehat{\ell(\theta)}_m := \frac{N}{m} \sum_{n=1}^{m} \log f(y_{i_n}|\theta)$$

where indices $(i_n)_{n=1}^m$ are uniformly drawn from $\{1, \ldots, N\}$, is an unbiased estimate of $\ell(\theta; y_1, \ldots, y_N)$. Moreover it is normally distributed when N and m go to infinity. In situations where the data set is very large (ie N is large), a MCMC scheme perhaps similar to the one proposed in [Ceperley 99] but relaxing the normality requirement, targeting the exact posterior distribution while only relying on log likelihood estimates computed on $m << N$ observations at each iteration of the algorithm, would of course be crucially useful.

Sequences of closely-related probability distributions

In this section, we consider a case of target distribution that is not necessarily more generic than the ones considered in the previous sections, but that has a particular sequential construction. Here the density function might be point-wise tractable like in Section 1.2.2, which will lead to the introduction of Sequential Monte Carlo samplers [Chopin 02, Del Moral 06], or only estimable like in Section 1.2.2, which will lead to random weight particle filters [Fearnhead 10] and SMC^2 (Chapter 6).

Here the target distribution is such that we can build a finite sequence of distributions satisfying the following: (1) we can draw a sample from the first distribution, (2) we can "transfer" a sample from one distribution to a sample from the next one, and (3) the final distribution of the sequence is precisely the target distribution. This is the common context in the Sequential Monte Carlo (SMC, or particle filters) literature. The first particle method was introduced in [Gordon 93] to approximate the sequence of filtering distributions $(p_\theta(x_t|y_{1:t}))_{t=1}^T$ introduced in Section 1.1.2, using the recursion formula, and under some assumptions on the transition and measurement densities. SMC methods in general are explained in details in the first chapters of [Doucet 01] along with improvements and applications in the following chapters; see also Chapters 7 and 8 of [Cappé 05]; and also [Cornebise 09], where the first chapters provide a clear and rigorous introduction. In the HMM context SMC methods are also called *particle filters*.

SMC methods were later extended to many cases, not necessarily related to time series. In this case they are usually called *SMC samplers*. Importantly, a generic posterior distribution in Bayesian statistics can naturally be seen as the endpoint of a sequence of distributions, defined as follows: given N observations, the sequence can be taken as $(\pi(\theta|y_{1:k}))_{k=0}^N$, using the notation of Section 1.1.1 and the convention $y_{1:0} = \varnothing$, $y_{1:k} = (y_1, \ldots y_k)$ for $k \geq 1$. In this natural construction the observations are added one by one, eventually leading to the posterior distribution given all the available observations. Other sequences are of course possible, for instance the sequence of tempered distributions $(\pi(\theta|y_{1:N})^{\gamma_k})_{k=0}^K$ where $(\gamma_k)_{k=0}^K$ is an increasing, positive real-valued sequence such that $\gamma_K = 1$. That sequence has the benefit of providing a smooth path between a very flat distribution, corresponding to a small value of γ_0, and the target distribution when $\gamma_K = 1$. In this context, referred to as a "static model" (as opposed to a "dynamic model" where the observations constitute time series), [Chopin 02] proposes the first proper SMC method (see also [Jarzynski 97] for an ancestor that [Neal 01] later brought to the statistical community), that sets itself in the case where the target pdf can be evaluated point-wise as in Section 1.2.2.

We have seen in Section 1.2.2 that some sampling methods produce Markov chains. SMC methods do not produce independent samples either, but instead they provide dependent, weighted samples (called the *particles*). More precisely particles constitute a collection of weighted values, denoted by $(x_i, w_i)_{i=1}^P$ where P is the number of particles, x_i is in $\mathcal{X}$, the state space on which the target distribution is defined, and w_i is the weight, that is, a real positive value. The literature sometimes refers to normalized weights, in which case the sum of the weights is equal to one. We say that the particles $(x_i, w_i)_{i=1}^P$ follow a distribution f if for any f-integrable function h, we have:

$$\frac{\sum_{i=1}^P w_i h(x_i)}{\sum_{i=1}^P w_i} \xrightarrow[P\to\infty]{a.s.} \mathbb{E}_f(h(X))$$

In the SMC framework, a CLT takes the following general form:

$$\sqrt{P}\left(\frac{\sum_{i=1}^P w_i h(x_i)}{\sum_{i=1}^P w_i} - \mathbb{E}_f(h(X))\right) \xrightarrow[P\to\infty]{Law} \mathcal{N}(0, \sigma^2(h))$$

for a positive real-value $\sigma^2(h)$ that does not depend on P. [Chopin 04] proves a CLT for SMC methods used in Bayesian inference, based on standard assumptions on the asymptotic behaviour of the likelihood function when the number of observations increases. Although more complex, most of the current research on the theoretical properties of SMC methods are based on Feynman–Kac representations introduced in [Del Moral 04] and applied to SMC for static problems in [Del Moral 06], where the authors find the same CLT as [Chopin 04]. Recently, new theoretical results have appeared in various directions: non-asymptotics [Cérou 11], under verifiable assumptions [Whiteley 11], when the dimension of the state space increases [Beskos 11a, Beskos 11b], etc.

When the target density is not tractable, there is a SMC counterpart to the MCMC methods presented in Section 1.2.2, introduced in [Fearnhead 10] and called Random Weight Particle Filter. Although the latter article sets itself in a continuous-time context, there is a discrete-time version where the ideal weights of the particles are replaced by unbiased, positive estimates of them. An instance of such an algorithm is the SMC^2 algorithm introduced in Chapter 6.

Totally intractable probability density function

In this last category, we mention a case where very little is known about the target distribution: its probability density function cannot be evaluated nor even estimated point-wise. In the Bayesian framework this is typically the case where the likelihood is intractable and no known unbiased estimate is available, so that even the generic methods presented in Section 1.2.2 are not applicable.

If we assume that we can simulate realizations from the likelihood, that is, from the statistical model given a parameter value, then we can apply the Approximate Bayesian Computation (ABC) method. This was introduced in [Pritchard 99] in a human evolution context, and benefits since then from a great surge in popularity in population genetics [Beaumont 02], dynamical systems [Toni 09] and many other fields. See the references in [Csilléry 10] for more applications and the references in [Barnes 11] for recent methodological improvements.

Similarly to MCMC and SMC methods, ABC provides a sample approximately distributed according to the posterior distribution of the parameters. There is a tuning parameter (denoted by ε), such that when it goes to 0, the sample becomes exactly sampled according to the posterior distribution. However the main concern with ABC is that the approximation error is much harder to control than the one associated with Monte Carlo methods like MCMC or even SMC, perhaps because its wide use is only recent; there are few theoretical results stating how well the approximation behaves when ε goes to 0; very recently [Fearnhead 12] look at the accuracy of the method in terms of estimation of certain functionals of the target, instead of using a distance between the generated sample and the target distribution.

To summarize, we can categorize the target distribution according to our knowledge about them, depending on which various Monte Carlo methods apply. For posterior distributions in the Bayesian framework, we provide a summary in Table 1.1. In this table the first column indicates the available knowledge on the posterior distribution, the second column indicates whether or not the posterior distribution is considered as the endpoint of a finite sequence of distributions as in Section 1.2.2, and the third column indicates the appropriate family of methods.

Posterior density	Sequence	Method
tractable	no	Metropolis–Hastings
	yes	SMC samplers
estimable	no	pseudo-marginal approach, PMCMC
	yes	Random Weight Particle Filter, SMC^2
otherwise	yes / no	ABC

Table 1.1: Generic Monte Carlo methods for Bayesian Statistics. This table summarizes the available Monte Carlo methods, depending on the properties of the posterior density function.

1.2.3 Using random samples to approximate integrals

We have seen in Section 1.1 that our objects of interest are functionals of target distributions, most often implying integration of some function with respect to the target density. Since integration is an ubiquitous problem in applied mathematics, one can legitimately wonder whether using random samples to approximate integrals is the best option available, and we will consider in this section the benefits of Monte Carlo compared to other options. Ideally one could try to actually get an explicit formula for the integral, bypassing any approximation. Alternatively one could resort to other approximation methods, as for instance the most well-known Simpson's rule.

Owen's tutorial on Monte Carlo methods [Owen 12] provides a clear exposition of the benefits of Monte Carlo and I will follow his arguments here. Let us first consider a case where the integral is challenging but could still be computed in closed-form. Though an uncommon case in Bayesian statistics, it can still shed some light on the benefits of Monte Carlo. Suppose we are interested in computing the average distance between two points uniformly drawn in a rectangle. [Mathai 99] presents the problem along with many references to applications. For $a, b > 0$ write $R = [0, a] \times [0, b]$ our rectangle, and consider random variables X and Z uniformly distributed on R. The average distance between X and Z is equal to the following:

$$\mu(a, b) = \frac{1}{a^2b^2} \int_0^a \int_0^b \int_0^a \int_0^b \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2} dx_1 dx_2 dz_1 dz_2$$

There happens to be a closed-form expression for this integral [Ghosh 51], allowing exact computation for any a and b. However the formula does not extend easily to other shapes than rectangles, nor does it extend easily to other distance than the L_2 distance used here. Indeed a considerable effort is required for each shape and norm, which might lead to a closed-form expression in the best case, and to nothing in other cases.

Using Monte Carlo, drawing uniform samples from a rectangle is straightforward, and hence the computation is similarly easy for any distance. To approximate the integral above, one can resort to the following Monte Carlo estimator:

$$\widehat{\mu(a,b)} = \frac{1}{N}\sum_{n=1}^{N}\sqrt{\left(X_1^{(n)} - Z_1^{(n)}\right)^2 + \left(X_2^{(n)} - Z_2^{(n)}\right)^2}$$

where

$$(X_1^{(n)}, X_2^{(n)})_{n=1}^{N} \sim \mathcal{U}\left(R\right) \text{ and } (Z_1^{(n)}, Z_2^{(n)})_{n=1}^{N} \sim \mathcal{U}\left(R\right)$$

In this formula $\mathcal{U}\left(R\right)$ denotes the uniform distribution on the rectangle R, and N is the number of generated points. Extensions to other shapes requires the ability to sample uniformly from the given shape, which is likely to be a much easier problem than computing a specific integral on that same shape. For a given shape S and a distance d, the approximation extends to:

$$\widehat{\mu(S)} = \frac{1}{N}\sum_{n=1}^{N} d(X^{(n)}, Z^{(n)})$$

where

$$(X^{(n)})_{n=1}^{N} \sim \mathcal{U}\left(S\right) \text{ and } (Z^{(n)})_{n=1}^{N} \sim \mathcal{U}\left(S\right)$$

As seen in the CLT given at the beginning of this section, the error (*e.g.* the Root Mean Square Error) of this estimate converges to 0 at the same speed as $1/\sqrt{N}$. This means that an arbitrary precision can be reached, but at a considerable cost: having one more significant digit requires a hundred times more sampled points. On the other hand, the error is controlled since estimates of the error are available given the generated sample. Using the bootstrap method, the estimates of the Monte Carlo error can actually converge faster (*e.g.* in $1/N$) than the estimate of the quantity of interest; see [Hall 86] in the iid case. This example illustrates that Monte Carlo methods can be used as an efficient and flexible method to get at least a first approximation of an integral of interest.

Let us now consider a case where the integral cannot be solved in closed-form, so the Monte Carlo method is now compared to alternative approximation methods. [Heinrich 00] provides an interesting review of results about optimal algorithms for integrating functions from Hölder or Sobolev spaces (see the article for definitions), first due to [Bakhvalov 59] and [Bakhvalov 62]. Essentially, if we consider the problem of computing $\mathcal{I}(\varphi)$ as in the beginning of the section, where φ is now in a regular space (a Hölder or Sobolev space, denoted by F), [Heinrich 00] defines the error of an approximation technique A as follows:

$$e(A,F) = \sup_{\varphi\in F}|\mathcal{I}(\varphi) - A(\varphi)|$$

where $A(\varphi)$ is an approximation of $\mathcal{I}(\varphi)$ provided by the method A. It is hence a worst-case scenario criterion to compare approximation methods. Then a variety of results describe the performance of the best algorithm (the one which provides the smallest error) among classes of algorithms (deterministic, randomized and quantum algorithms). To summarize the results, it is found that randomized algorithms yield better performance than their deterministic counterpart, and that the Monte Carlo method provides a nearly optimal convergence rate (again, for this worst-case scenario criterion) when the dimension is large or when the smoothness of the integrand is low. More precisely, define a Sobolev space $W_{p,d}^k$ as follows:

$$W_{p,d}^k = \{f : \|D^i f\|_p \leq 1, |i| \leq k\}$$

where d denotes the dimension of the state space, i is a multiindex and $|i|$ is the sum of its components. The optimal randomized algorithm based on N values to integrate functions in $W^k_{p,d}$ has an error as follows:

$$e_N^{\mathrm{ran}}(W^k_{p,d}) := \inf_{A_N \in \mathcal{A}^{\mathrm{ran}}} e(A_N, W^k_{p,d}) \approx N^{-k/d-1/2}$$

for $2 \leq p \leq \infty$, where $\mathcal{A}^{\mathrm{ran}}$ is the class of randomized algorithms, and the $\approx$ sign means the following: $a_N \approx b_N$ if there are $c_1, c_2 > 0$ and $N_0 \in \mathbb{N}$ such that $N \geq N_0$ implies $c_1 b_N \leq a_N \leq c_2 b_N$.

Compared to the basic Monte Carlo error of order $1/\sqrt{N}$, we see that whenever the ratio k/d is small, then Monte Carlo approximations are nearly optimal. We are here dismissing quantum algorithms presented in [Heinrich 00], whose current applicability is rare, if promising. Obviously the basic Monte Carlo method is not applicable in every case: as we have seen in Section 1.2.2, we will not have an independently distributed sample from the target distribution using MCMC or SMC methods. However the approximation will still deteriorate with the dimension of the sample space, for a fixed sample size. This amounts to observe that, if the rate of the Monte Carlo method is always $1/\sqrt{N}$, the dimension d will still appear in a constant that plays a crucial role in practice, making the theoretical rates in N, as described above, somewhat less relevant. More appropriate results as in [Roberts 01] explain how the Metropolis–Hastings behaves when the dimension of the state space increases, and how the algorithm should be tuned accordingly. [Beskos 11b] and [Beskos 11a] investigate the same effect on SMC methods.

1.3 Challenging aspects of some target distributions

We have seen that a plethora of Monte Carlo methods allows to sample from various target distributions, even if little is known about them as in Section 1.2.2. However even if methods are available, they do not perform equally well in every situation. Among the numerous issues that can make the computation challenging, we will pay particular attention to two classical issues: computational cost and multimodality.

1.3.1 Expensive target density evaluation

Consider cases where the target density can be evaluated point-wise as in Section 1.2.2, but each evaluation is very costly. This often happens in Bayesian statistics, since the posterior density evaluation involves an exact evaluation of the likelihood, for which the computational cost is directly linked with the sample size; at least if no sufficient statistics are available. [Wraith 09] provides references to examples in cosmology where each evaluation of the likelihood function takes at least several seconds, even using very optimized code (CAMB, see [Lewis 00]), partly because the data set is large and partly because the models involve a lot of computation for each piece of data.

In this situation, Monte Carlo methods are applicable; however they will only generate small samples, in a reasonable time frame, and hence the resulting estimation might be imprecise. For instance, if each target density evaluation takes about 5 seconds, it would take several days to produce a sample of size 10^5 using the MH algorithm. In the next sections we will describe two research directions aiming at improving the estimation, for a fixed time horizon. The first one, Rao–Blackwellization, can be seen as a way to make the computation more precise for a given sample size, while the second one, parallel computing, can be seen as a way to produce a larger sample without increasing the human computing time.

Rao–Blackwellization

In the original MH setting, every time that a proposal is rejected, the associated target density value is thrown away, even though it contains some information on the target distribution. As this clearly seems suboptimal, methods have appeared to recycle the rejected proposals and their associated target density values.

Chapter 2 presents such a method called Block IMH, tightly linked with two previous articles [Casella 96, Douc 11]. It is not exactly a new MCMC algorithm, but rather a method to post-process the chain generated by a Metropolis–Hastings using an independent proposal distribution, in order to reduce the variance of the resulting estimates. Hence using this method, less iterations have to be performed to obtain a given precision in the estimate than with the original method. It therefore falls into the category of variance reduction techniques (see Chapter 4 of [Robert 04] for a general presentation of these techniques), which aim at diminishing the variance of estimates for a fixed computational cost.

The variance reduction in the Block IMH algorithm comes from a Rao–Blackwellization argument, which can be summarized as follows. Using the notation of the beginning of the section, consider again our Monte Carlo estimate

$$\widehat{\mathcal{I}(\varphi)}^N := \frac{1}{N}\sum_{n=1}^{N} h(X_n)$$

Then for any random variable Y we have:

$$\mathbb{V}\left[\mathbb{E}\left(\widehat{\mathcal{I}(\varphi)}^N \middle| Y\right)\right] \leq \mathbb{V}\left(\widehat{\mathcal{I}(\varphi)}^N\right)$$

and of course:

$$\mathbb{E}\left[\mathbb{E}\left(\widehat{\mathcal{I}(\varphi)}^N \middle| Y\right)\right] = \mathbb{E}\left(\widehat{\mathcal{I}(\varphi)}^N\right) = \mathcal{I}(\varphi)$$

which implies that if we find Y such that we can easily compute the expectation of the Monte Carlo estimate conditional upon Y then we should use this estimate instead of the original one, since the expectation is the same and the variance is necessarily smaller or equal.

Using Rao–Blackwellization techniques in practice amounts to finding an appropriate variable Y: one that both significantly decreases the variance and does not add a lot of extra computation; otherwise the benefit might be null compared to simply using the original Monte Carlo estimate with more samples. The Block IMH algorithm has the benefit of being designed for parallel computing, and hence its additional cost compared to standard MH is in practice very little. The following section provides an introduction to parallel computing in the statistical computing context.

Parallel Computing

Parallel computing currently constitutes a popular solution to expensive computational costs, and beside Block IMH (Chapter 2), PAWL (Chapter 5) and SMC2 (Chapter 6) have also been designed to take this new computational paradigm into account.

In a standard computer the instructions are carried out by the Central Processing Unit (CPU). Over the years, CPUs have become faster and faster due to constant progress in miniaturization, and they are generally assumed to follow Moore's law that states that their speed is doubling every two years or so. This impressive rate implies that computers are, in average, expected to be 1000 times faster in 20 years. Another obvious way to aim at a 1000 times speedup in computations is to use 1000 computers. At the time of this writing the fastest supercomputer in the world (Fujitsu K Computer in Kobe, Japan) combines up to $88,128$ processors. Supercomputers emerged in the 1960s and proved useful for a variety of computationally intensive applications, but are not easily accessible and their cost is at least in the millions: with so many processors at the same location, special infrastructures have to be built to handle the heat and the huge electrical consumption, among other issues. The sole cost of cooling the system can be about several million dollars a year.

Recently, the scientific community discovered a much cheaper mean to obtain similar speedups using Graphics Processing Units (GPUs), that originally equipped gaming platforms such as high-end personal computers and consoles. For a similar price as a standard CPU, a GPU contains hundreds of cores, which act as independent processing units each capable of performing a task at the same time. Additionally GPUs can easily be stacked in small clusters, so that it is now within every laboratory's budget to build a small device containing a few thousands cores, with very low electrical consumption. Performing general taks on a GPU is usually referred to as General-Purpose computing on GPUs (GPGPU). In the 1990s this was known as a very tedious task, since the GPU was designed only to handle graphical objects. The programmer had to trick the GPU into performing his computation by expressing his data as "textures" and his functions as "shaders". GPGPU has widely spread in the 2000s thanks to a lot of publicly available libraries such as OpenCL and CUDA, that made the implementation on GPU similar to standard programming; GPUs are now also designed for other purposes than displaying objects, some brands directly targeting the supercomputing market.

Hence, access to parallel computing has become very easy. Some algorithms can greatly benefit from such architectures while others cannot. Tasks that can benefit from parallel cores are called "parallelizable". A typical parallelizable task is the evaluation of the same function on each component of a large vector. A typical non-parallelizable task is an iterative algorithm where the result of iteration i is needed to start iteration $i+1$. Hence, a generic MCMC algorithm (such as the Random Walk Metropolis–Hastings) does not in general benefit from parallelization: it is an iterative algorithm generating a Markov chain, and each iteration requires the result of the previous one. The MH algorithm with independent proposals is an exception where most of the computation can be parallelized. SMC methods are, on the other side, much better suited for parallel implementation, as was noted *e.g.* in [Holmes 11] who notices speedups up to 500 fold using a standard GPU compared to a standard CPU. Indeed, most of the computation in a SMC algorithm can be done independently for each particle, with the exception of the resampling step; and current efforts [Murray 12, Ahmed 11] aim at dealing with this bottleneck. Even when the algorithm is purely iterative, significant speedups can be seen if each iteration involve a huge computation, but that is entirely model dependent (see *e.g.* [Suchard 10] for an example in mixture models).

To summarize, a new interest has risen regarding "parallel-friendly" algorithm, where all the parallel cores can be fully exploited. MCMC algorithms using parallel chains or SMC algorithms hence have a new advantage over single chain MCMC algorithms.

1.3.2 Multimodality in target distributions

A well-known difficulty occurs when the target distribution is multimodal (or polymodal), which means that its density presents multiple modes (maxima and minima). In this setting many sampling methods exhibit poor performance. For instance the chains generated by the MH algorithm tend to have difficulties to exit the first local mode they encounter, and then fail to recover the other modes. Indeed, in the MH algorithm, proposed points with lower density values than the current point are not likely to be accepted; but to escape from a local mode, the chain has to cross a region of the state space associated with low density values. While this crossing is going to happen eventually (as granted by the theoretical convergence of the MH algorithm), in practice this can take a prohibitive number of iterations. This is obviously a concern, since it results in a bias on the resulting computation based on the chain.

Chapters 3, 4 and 5 consider algorithms (namely the Free Energy SMC algorithm and variations of the Wang–Landau algorithm) that are designed to tackle this issue. SMC in general, provided that the initial distribution of the chosen sequence of distributions is flat enough, is considered to be more robust than MCMC when the target distribution is multimodal, as already noted in [Chopin 02]. Hence the ability to tackle multimodality in the target is also an advantage of SMC^2 (Chapter 6) over PMCMC methods [Andrieu 10]; and thus a central motivation of this thesis.

We first introduce the mixture model as an example where the posterior distribution is notoriously multimodal. This example is also used for illustrative purposes in Chapters 3 and 5.

Multimodality in mixture models

Let us first define a mixture model and exhibit the posterior distribution. We will focus on mixtures of normal distributions, pointing to [Frühwirth-Schnatter 06] for a full introduction to many other mixture models in the Bayesian framework; see also [Lee 08] for a recent survey. We will also consider the number of components K to be known; Monte Carlo methods to tackle the case where K is unknown are described *e.g.* in [Green 95, Stephens 00a, Jasra 08]. A N-sample $(Y_1, \dots, Y_N)$ is said to follow a normal mixture distribution with K components if each random variable Y_n is independent from the others and follows a distribution with a density f defined as:

$$f(y) = \sum_{k=1}^{K} \omega_k \varphi(y; \mu_k, \sigma_k^2)$$

where $\varphi(\cdot; \mu, \sigma^2)$ denotes the density of a normal distribution with mean μ and variance σ^2, and $(\omega_k)_{k=1}^K \in [0,1]^K$ (called the weights) are such that $\sum_{k=1}^K \omega_k = 1$; hence f is obviously a probability density function. The parameters to estimate are in general $(\omega_k, \mu_k, \sigma_k^2)_{k=1}^K$, noting that only the first $K-1$ weights have to be estimated, the last one being necessarily equal to $1 - \sum_{k=1}^{K-1} \omega_k$. We will denote the parameter to estimate by θ; it lives in a subset Θ of $\mathbb{R}^{3K-1}$. Denoting by p the prior density assigned to the parameter θ, the posterior density is as follows:

$$\pi(\theta | y_1, \dots, y_N) \propto p(\theta) \prod_{n=1}^{N} \sum_{k=1}^{K} \omega_k \varphi(y_n; \mu_k, \sigma_k^2)$$

[Celeux 00] provides a good introduction to the difficulties of sampling according to the posterior distribution. First, this posterior distribution exhibits the so-called “label switching” issue: for a given parameter value any permutation of the components leads to the same likelihood value and hence to the same posterior density value. More formally,

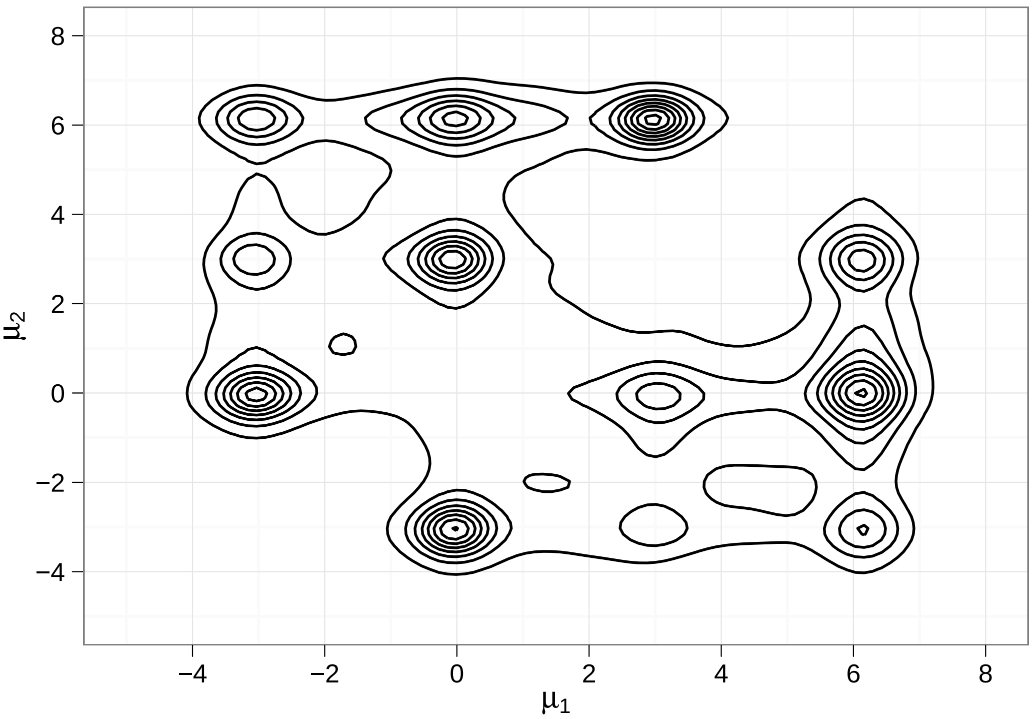


Figure 1.1: Posterior distribution of two parameters (μ_1, μ_2) in a mixture model with 4 components and 100 data points generated from it. The contours are recovered using a sample from the posterior distribution, obtained using the parallel Wang–Landau algorithm. Details are given in Chapter 5.

for a given parameter value:

$$\theta = (\omega_1, \omega_2, \dots, \omega_{K-1}, \mu_1, \mu_2, \dots, \mu_K, \sigma_1^2, \sigma_2^2, \dots, \sigma_K^2)$$

if we define for example the following permutation of the components:

$$\tilde{\theta} = (\omega_2, \omega_1, \dots, \omega_{K-1}, \mu_2, \mu_1, \dots, \mu_K, \sigma_2^2, \sigma_1^2, \dots, \sigma_K^2)$$

then $\pi(\theta|y_1, \dots, y_N) = \pi(\tilde{\theta}|y_1, \dots, y_N)$. Alternatively one can set constraints on the parameter space, for instance by enforcing $\mu_1 < \mu_2 < \ldots < \mu_K$, which makes the statistical model identifiable; although [Stephens 00b] explains why this does not actually solve the issue since the posterior distribution can still be multimodal. The multimodality is illustrated in Figure 1.1, showing the marginal posterior distribution of (μ_1, μ_2), obtained by using the parallel Wang–Landau algorithm described in Chapter 5. More precisely, a synthetic data set is first drawn from the model for a fixed parameter value, and a prior distribution is defined on the parameters as *e.g.* in [Richardson 97]. Then the algorithm generates a sample, from which a kernel density estimate can be computed and shown in the figure.

With Monte Carlo methods in mind, a reason for not imposing this identifiability constraint is precisely to keep the multimodality intact for testing purposes. This multimodality has the benefit of being predictable: we know that each mode of the posterior density in the constrained space has $K!$ copies (the number of label permutations) in the unconstrained space; hence if a method produces a sample showing less than $K!$ modes in total then it necessarily missed some huge part of the state space. In the MCMC framework, we will be able to conclude immediately that the method did not run for enough iterations. For example, in Figure 1.1 an ideal algorithm would have recovered $(K-1)!$ identical modes; and by looking at the other marginal distributions: (μ_1, μ_3), (μ_2, μ_3) and so on, one could count whether or not (at least) $K!$ modes were explored. Note also that mixture models exhibit more modes than those coming from the label switching phenomenon; it also exhibits multiple "genuine" modes. [Stephens 00b] shows an example of "genuine" multimodality, more of such examples can be found in the PhD thesis of the same author [Stephens 97].

Therefore in this document, and especially in Chapters 3 and 5, we will consider the posterior distribution of normal mixture models without any identifiability constraint as a benchmark for sampling methods, in order to assess their ability to recover many distinct modes in the target density. This model is thus often used to illustrate sampling methods expected to deal with multimodality in the target distribution, including SMC [Chopin 02, Jasra 07], Free Energy MCMC [Chopin 11], Importance Tempering [Gramacy 10], adaptive MCMC [Atchadé 11], Parallel Tempering with Equi-Energy Moves [Baragatti 11] among others.

Multimodality in Hidden Markov models

Nothing prevents multimodality from occurring in the challenging HMM context. Multimodality in the filtering distribution was already illustrated in the seminal paper on Sequential Monte Carlo [Gordon 93]. Multimodality in the posterior distribution of the parameter presents a particular difficulty since few efficient methods are able to sample from this target distribution in general, even when it does not present multimodality. Still, if the likelihood function is multimodal, then the posterior distribution is expected to be multimodal as well and that was reported in several situations, including an interesting example in population ecology described in [Polansky 09]. Let us introduce the context of this paper, which models the per capita growth rate of some species like *Accipiter nisus* (also known as the Eurasian Sparrowhawk), a bird of prey.

We observe T values $(Y_1, \ldots, Y_T) \in \mathbb{R}^T$ representing population densities of some species. We assume that these are noisy measurements of the "true" population density, denoted by $(N_t)_{t=1}^T$. Furthermore we put a model on this hidden population density as follows:

$$\log N_{t+1} = \log N_t + g(N_t) + \sigma_p v_t$$

where $(v_t)_{t=1}^T$ are independent standard normal variables, $\sigma_p > 0$, and g is a function that takes different forms depending on the model. In the particular case of the so-called "theta-Ricker" growth model, g takes the following form:

$$g(N_t) = r\left(1 - \left(\frac{N_t}{K}\right)^\theta\right)$$

where r is the net per capita growth rate and K is the carrying capacity, *i.e.* the population size at equilibrium. The parameter θ defines the form of the dependence between the density and its growth rate. A clear introduction to these models is provided in [Pastor 11] (see in particular Chapter 6 for the Ricker model). The observations are noisy measurements of N_t:

$$Y_t = N_t + \sigma_o \varepsilon_t$$

where $(\varepsilon_t)_{t=1}^T$ are independent standard normal variables and $\sigma_o > 0$. This measurement function seems problematic since Y_t can be negative, although it should represent population densities. However in practice σ_o is very small compared to $(N_t)_{t=1}^T$ and hence this measurement function is commonly used in this simple form.

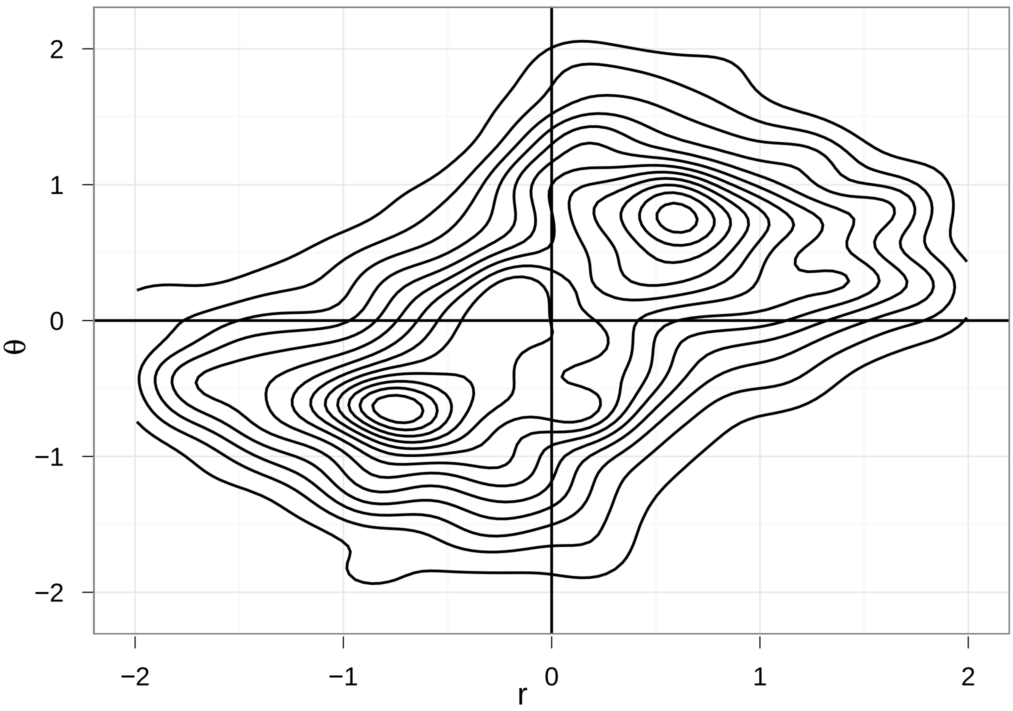


Figure 1.2: Marginal posterior distribution of (r, θ) in the theta-Ricker model applied to the *Accipiter nisus* data set, obtained using SMC^2 with $N_\theta = 5,000$ θ-particles. This distribution shows a clear bimodality, as discussed in [Polansky 09]. Details of the sampling method are given in Chapter 6.

We are interested in the posterior distribution of the parameters, which are here the noise variances (σ_p^2 and σ_o^2), the initial value of the hidden process (N_0), and the parameters specific to the theta-Ricker model (r, K and θ). We formulate the following basic prior distribution:

$$\begin{aligned}
\sigma_o^2, \sigma_p^2 &\sim \mathcal{IG}(2,1)\\
\log N_0 &\sim \mathcal{N}(\log \bar{Y}, 1)\\
r &\sim \mathcal{N}(0,1)\\
K &\sim \mathcal{N}_+(\bar{Y}, 1)\\
\theta &\sim \mathcal{N}(0,1)
\end{aligned}$$

where $\mathcal{IG}$ is the inverse-Gamma distribution, $\mathcal{N}_+$ is the normal distribution truncated on $(0, +\infty)$, and $\bar{Y}$ is the mean of the observations.

Our data set of population densities of sparrowhawks is made of 18 values. To get a representation of the posterior density of the parameters we launch the SMC^2 algorithm, explained at length in Chapter 6. We obtain a sample distributed according to the posterior distribution of the parameters, and we are interested in the marginal distribution of (r, θ). We plot a kernel density estimate of its density in Figure 1.2, where it is pretty clear that the distribution is at least bimodal. [Polansky 09] mentions that biological rationale can lead to force r and θ to be positive. Indeed if these are negative, the growth rate goes to infinity when the population size goes to 0, which of course is not a desirable property. Even with these constraints, [Polansky 09] reports multimodality in the likelihood on simulated data sets, as well as a ridges in the likelihood function.

Bibliography

[Ahmed 11] M. Ahmed, P. Bibalan, N. de Freitas & S. Fauvel. *Decentralized, Adaptive, Look-Ahead Particle Filtering.* IEEE Proceedings on Signal Processing, vol. 59, no. 2, 2011.

[Andrieu 09] C. Andrieu & G.O. Roberts. *The pseudo-marginal approach for efficient Monte Carlo computations.* Ann. Stat., vol. 37, no. 2, pages 697–725, 2009.

[Andrieu 10] C. Andrieu, A. Doucet & R. Holenstein. *Particle Markov chain Monte Carlo (with discussion).* J. Royal Statist. Society Series B, vol. 72, no. 4, pages 357–385, 2010.

[Atchadé 11] Y. Atchadé, G. Fort, E. Moulines & P. Priouret. *Adaptive Markov chain Monte Carlo : theory and methods.* In D. Barber, A. T. Cemgil & S. Chiappia, editeurs, Bayesian Time Series Models, pages 33–52. Cambridge Univ. Press, 2011.

[Bach 12] F. Bach, S. Lacoste-Julien & G. Obozinski. *On the Equivalence between Herding and Conditional Gradient Algorithms.* In Proceedings of the 29th Annual International Conference on Machine Learning, ICML '12, New York, NY, USA, 2012. ACM.

[Bakhvalov 59] N.S. Bakhvalov. *On approximate computation of integrals.* Vestnik Moskov. Gos. Univ. Ser. Math. Mech. Astron. Phys. Chem., vol. 4, pages 3–18, 1959.

[Bakhvalov 62] N.S. Bakhvalov. *On the rate of convergence of indeterministic integration processes within the functional classes $W_p^{(l)}$.* Theory Prob. Applications, vol. 7, page 227, 1962.

[Baragatti 11] M. Baragatti, A. Grimaud & D. Pommeret. *Parallel tempering with equi-energy moves.* ArXiv e-prints, January 2011.

[Barnes 11] C. Barnes, S. Filippi, M. P. H. Stumpf & T. Thorne. *Considerate approaches to achieving sufficiency for ABC model selection.* ArXiv e-prints, June 2011.

[Bauwens 11] L. Bauwens, A. Dufays & J. V.K. Rombouts. *Marginal likelihood for Markov switching and change-point GARCH models.* CREATES Research Papers 2011-41, School of Economics and Management, University of Aarhus, November 2011.

[Beaumont 02] M.A. Beaumont, W. Zhang & D.J. Balding. *Approximate Bayesian computation in population genetics.* Genetics, vol. 162, no. 4, page 2025, 2002.

[Beaumont 03] M.A. Beaumont. *Estimation of population growth or decline in genetically monitored populations.* Genetics, vol. 164, no. 3, pages 1139–1160, 2003.

[Beskos 11a] A. Beskos, D. Crisan & A. Jasra. *On the stability of sequential Monte Carlo methods in high dimensions.* ArXiv e-prints, March 2011.

[Beskos 11b] A. Beskos, D. Crisan, A. Jasra & N. Whiteley. *Error bounds and normalizing constants for sequential Monte Carlo in high dimensions.* ArXiv e-prints, December 2011.

[Cappé 05] O. Cappé, E. Moulines & T. Rydén. Inference in hidden Markov models. Springer-Verlag, New York, 2005.

[Casella 96] G. Casella & C.P. Robert. *Rao-Blackwellisation of sampling schemes.* Biometrika, vol. 83, no. 1, pages 81–94, 1996.

[Casella 01] G. Casella, M. Lavine & C.P. Robert. *Explaining the perfect sampler.* The American Statistician, vol. 55, no. 4, pages 299–305, 2001.

[Celeux 00] G. Celeux, M.A. Hurn & C.P. Robert. *Computational and inferential difficulties with mixtures posterior distribution.* J. American Statist. Assoc., vol. 95(3), pages 957–979, 2000.

[Ceperley 99] D. M. Ceperley & M. Dewing. *The penalty method for random walks with uncertain energies.* J. Chem. Phys., vol. 20, no. 110, pages 9812–9821, 1999.

[Cérou 11] F. Cérou, P. Del Moral & A. Guyader. *A nonasymptotic theorem for unnormalized Feynman–Kac particle models.* Ann. Inst. Henri Poincarré, vol. 47, no. 3, pages 629–649, 2011.

[Chen 10] Y. Chen, M. Welling & A. J. Smola. *Super-Samples from Kernel Herding.* In Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, UAI '10, 2010.

[Chopin 02] N. Chopin. *A sequential particle filter method for static models.* Biometrika, vol. 89, pages 539–552, 2002.

[Chopin 04] N. Chopin. *Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference.* Ann. Statist., vol. 32, no. 6, pages 2385–2411, 2004.

[Chopin 11] N. Chopin, T. Lelievre & G. Stoltz. *Free energy methods for Bayesian inference: efficient exploration of univariate gaussian mixture posteriors.* Statistics and Computing, page 20 p., 2011.

[Cornebise 09] J. Cornebise. *Adaptive Sequential Monte Carlo methods.* PhD thesis, Université Pierre et Marie Curie, Paris, 2009.

[Csilléry 10] K. Csilléry, M. G.B. Blum, O.E. Gaggiotti & O. François. *Approximate Bayesian Computation (ABC) in practice.* Trends in Ecology & Evolution, vol. 25, no. 7, pages 410 – 418, 2010.

[Del Moral 04] P Del Moral. Feynman-Kac formulae. Springer, 2004.

[Del Moral 06] P. Del Moral, A. Doucet & A. Jasra. *Sequential Monte Carlo samplers.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 68, no. 3, pages 411–436, 2006.

[Devroye 85] L. Devroye. Non-uniform random variate generation. Springer-Verlag, New York, 1985.

[Douc 11] R. Douc & C.P. Robert. *A vanilla Rao-Blackwellization of Metropolis-Hastings algorithms.* Ann. Stat., vol. 39, no. 1, pages 261–277, 2011.

[Doucet 01] A. Doucet, N. de Freitas & N. Gordon. Sequential Monte Carlo methods in practice. Springer-Verlag, New York, 2001.

[Fearnhead 10] P. Fearnhead, O. Papaspiliopoulos, G.O. Roberts & A. Stuart. *Random weight particle filtering of continuous time processes.* J. R. Stat. Soc. Ser. B Stat. Methodol., vol. 72, pages 497–513, 2010.

[Fearnhead 12] P. Fearnhead & D. Prangle. *Constructing summary statistics for approximate Bayesian computation: semi-automatic abc (with discussion).* J. R. Stat. Soc. Ser. B Stat. Methodol. (to appear), 2012.

[Frühwirth-Schnatter 06] S. Frühwirth-Schnatter. Finite mixture and Markov switching models. Springer-Verlag, New York, New York, 2006.

[Gelman 10] A. Gelman, S. Brooks, G. Jones & X.L. Meng. Handbook of Markov chain Monte Carlo: methods and applications. Chapman & Hall/CRC handbooks of modern statistical methods. CRC Press, 2010.

[Ghosh 51] B. Ghosh. *Random distances within a rectangle and between two rectangles.* Bull. Calcutta Math. Soc., vol. 43, pages 17–24, 1951.

[Golightly 11] A. Golightly & D.J. Wilkinson. *Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo.* Interface Focus, vol. 1, no. 6, pages 807–820, 2011.

[Gordon 93] N. Gordon, J. Salmond & A.F.M. Smith. *A novel approach to non-linear/non-Gaussian Bayesian state estimation.* IEEE Proceedings on Radar and Signal Processing, vol. 140, pages 107–113, 1993.

[Gramacy 10] R. Gramacy, R. Samworth & R. King. *Importance tempering.* Statistics and Computing, vol. 20, pages 1–7, 2010. 10.1007/s11222-008-9108-5.

[Green 95] P.J. Green. *Reversible jump MCMC computation and Bayesian model determination.* Biometrika, vol. 82, no. 4, pages 711–732, 1995.

[Hall 86] P. Hall. *On the number of bootstrap simulations required to construct a confidence interval.* Ann. Stat., vol. 14, pages 1453–1462, 1986.

[Hastings 70] W.K. Hastings. *Monte Carlo sampling methods using Markov chains and their application.* Biometrika, vol. 57, pages 97–109, 1970.

[Heinrich 00] S. Heinrich & E. Novak. *Optimal summation and integration by deterministic, randomized, and quantum algorithms.* In K.T. Fang, F.J. Hickernell & H. Niederreiter, editeurs, Monte Carlo and Quasi-Monte Carlo Methods, pages 50–62. Springer, Berlin, 2000.

[Holmes 11] C.C. Holmes, A. Doucet, A. Lee, M. Giles & C. Yau. *Bayesian computation on graphics cards.* In J.M Bernardo, M.J. Bayarri, J.O. Degroot, A.P. Dawid, D. Heckerman, A.M. Smith & M. West, editeurs, Bayesian Statistics 9: Proceedings of the Ninth Valencia International Meeting, June 3-8, 2010. Oxford University Press, 2011.

[Jarzynski 97] C. Jarzynski. *Nonequilibrium Equality for Free Energy Differences.* Phys. Rev. Lett., vol. 78, pages 2690–2693, Apr 1997.

[Jasra 07] A. Jasra, D.A. Stephens & C.C. Holmes. *On population-based simulation for static inference.* Statistics and Computing, vol. 17, no. 3, pages 263–279, 2007.

[Jasra 08] A. Jasra, A. Doucet, D.A. Stephens & C.C. Holmes. *Interacting sequential Monte Carlo samplers for trans-dimensional simulation.* Computational Statistics and Data Analysis, vol. 52, no. 4, pages 1765 – 1791, 2008.

[Kalman 61] R. E. Kalman & R. S. Bucy. *New results in linear filtering and prediction theory.* Trans. Amer. Soc. Mech. Eng., J. Basic Eng., vol. 83, pages 95–108, 1961.

[L'Ecuyer 01] P. L'Ecuyer, R. Simard, E. Jack Chen & W. David Kelton. *An object-oriented random-number package with many long streams and substreams.* Operations Research, vol. 50, pages 1073–1075, 2001.

[Lee 08] K. Lee, J.-M. Marin, K. Mengersen & C. P. Robert. *Bayesian inference on mixtures of distributions.* ArXiv e-prints, April 2008.

[Lewis 00] A. Lewis, A. Challinor & A. Lasenby. *Efficient computation of cosmic microwave background anisotropies in closed friedmann-robertson-walker models.* The Astrophysical Journal, vol. 538, pages 473–476, 2000.

[Marsaglia 03] G. Marsaglia. *Xorshift RNGs.* Journal Of Statistical Software, vol. 8, no. 14, pages 1–6, 2003.

[Mathai 99] A. Mathai, P. Moschopoulos & G. Pederzoli. *Random points associated with rectangles.* Rendiconti del Circolo Matematico di Palermo, vol. 48, pages 163–190, 1999. 10.1007/BF02844387.

[Metropolis 53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller & E. Teller. *Equations of state calculations by fast computing machines.* J. Chem. Phys., vol. 21, no. 6, pages 1087–1092, 1953.

[Meyn 93] S.P. Meyn & R.L. Tweedie. Markov chains and stochastic stability. Springer-Verlag, New York, 1993.

[Murray 12] L. Murray. *GPU acceleration of the particle filter: the Metropolis resampler.* ArXiv e-prints, February 2012.

[Neal 01] R. M. Neal. *Annealed importance sampling.* Statist. Comp., vol. 11, pages 125–139, 2001.

[Nicholls 12] G. Nicholls. *Metropolis–Hastings with randomized acceptance rates.* Confronting Intractability Workshop, Bristol, 2012.

[Nummelin 02] E. Nummelin. *MC's for MCMC'ists.* International Statistical Review / Revue Internationale de Statistique, vol. 70, no. 2, pages pp. 215–240, 2002.

[Owen 12] A. Owen. *Monte Carlo ideas and methods.* Tutorial at the Monte Carlo and Quasi-Monte Carlo Methods conference, 2012.

[Pastor 11] J. Pastor. Mathematical ecology of populations and ecosystems. John Wiley and Sons, 2011.

[Peters 10] G. W. Peters, G. R. Hosack & K. R. Hayes. *Ecological non-linear state space model selection via adaptive particle Markov chain Monte Carlo (AdPMCMC).* ArXiv e-prints, May 2010.

[Peters 11] G. W. Peters, M. Briers, P. V. Shevchenko & A. Doucet. *Calibration and filtering for multi factor commodity models with seasonality: incorporating panel data from futures contracts.* ArXiv e-prints, May 2011.

[Polansky 09] L. Polansky, P. de Valpine, J.O. Lloyd-Smith & W.M. Getz. *Likelihood ridges and multimodality in population growth rate models.* Ecology, vol. 90, no. 8, pages 2313–2320, 2009.

[Pritchard 99] J.K. Pritchard, M.T. Seielstad, A. Perez-Lezaun & M.W. Feldman. *Population growth of human Y chromosomes: a study of Y chromosome microsatellites.* Molecular Biology and Evolution, vol. 16, no. 12, page 1791, 1999.

[Propp 96] J.G. Propp & D.B. Wilson. *Exact sampling with coupled Markov chains and applications to statistical mechanics.* Random Structures and Algorithms, vol. 9, pages 223–252, 1996.

[Rasmussen 11] D.A. Rasmussen, O. Ratmann & K. Koelle. *Inference for nonlinear epidemiological models using genealogies and time series.* PLoS Comput Biol, vol. 7, no. 8, page e1002136, 08 2011.

[Richardson 97] S. Richardson & P. J. Green. *On Bayesian Analysis of Mixtures with an Unknown Number of Components (with discussion).* Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 59, no. 4, pages 731–792, 1997.

[Robert 94] C.P. Robert. The Bayesian choice. Springer-Verlag, New York, 1994.

[Robert 04] C.P. Robert & G. Casella. Monte Carlo statistical methods. Springer-Verlag, New York, second edition, 2004.

[Roberts 01] G.O. Roberts & J.S. Rosenthal. *Optimal scaling for various Metropolis-Hastings algorithms.* Stat. Sci., vol. 16, no. 4, pages 351–367, 2001.

[Stephens 97] M. Stephens. *Bayesian methods for mixtures of normal distributions.* PhD thesis, University of Oxford, 1997.

[Stephens 00a] M. Stephens. *Bayesian analysis of mixture models with an unknown number of components—an alternative to reversible jump methods.* Ann. Statist., vol. 28, pages 40–74, 2000.

[Stephens 00b] M. Stephens. *Dealing with label switching in mixture models.* J. Royal Statist. Society Series B, vol. 62(4), pages 795–809, 2000.

[Suchard 10] M.A. Suchard, Q. Wang, C. Chan, J. Frelinger, A. Cron & M. West. *Understanding GPU programming for statistical computation: studies in massively parallel massive mixtures.* Journal of Computational and Graphical Statistics, vol. 19, no. 2, pages 419–438, 2010.

[Tierney 94] L. Tierney. *Markov chains for exploring posterior distributions (with discussion).* Ann. Statist., vol. 22, pages 1701–1786, 1994.

[Toni 09] T. Toni, D. Welch, N. Strelkowa, A. Ipsen & M.P.H. Stumpf. *Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems.* Journal of the Royal Society Interface, vol. 6, no. 31, page 187, 2009.

[Von Neumann 51] J. Von Neumann. *Various techniques used in connection with random digits.* J. Resources of the National Bureau of Standards–Applied Mathematics Series, vol. 12, pages 36–38, 1951.

[Welling 09] M. Welling. *Herding dynamical weights to learn.* In Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, pages 1121–1128, New York, NY, USA, 2009. ACM.

[Whiteley 11] N. Whiteley. *Stability properties of some particle filters.* ArXiv e-prints, September 2011.

[Wraith 09] D. Wraith, M. Kilbinger, K. Benabed, O. Cappé, J.-F. Cardoso, G. Fort, S. Prunet & C.P. Robert. *Estimation of cosmological parameters using adaptive importance sampling.* Physical Review D, vol. 80, page 023507, 2009.

Chapter 2

Block Independent Metropolis–Hastings algorithm

Ce chapitre présente une méthode pour réduire la variance de l'estimation fondée sur une chaîne générée par l'algorithme de Metropolis–Hastings à proposition indépendante. Dans le contexte de l'émergence de l'utilisation du calcul parallèle en statistique, cet algorithme fait partie des méthodes très facilement parallélisables, ce qui n'est pas le cas de sa contrepartie utilisant une proposition à marche aléatoire.

Tout en gardant la structure Markovienne de l'algorithme, il est possible de réduire la variance de l'algorithme en utilisant des techniques de Rao–Blackwellisation. D'une part, puisque les propositions sont tirées suivant une même loi à chaque itération de l'algorithme, l'ordre dans lequel elles sont fournies à l'algorithme ne devrait pas avoir d'impact sur les résultats de l'inférence. Il est également possible diminuer partiellement le bruit provenant du tirage de variables aléatoires uniformes servant à accepter ou rejeter les propositions.

La méthode proposée dans ce chapitre combine ces deux effets: les itérations de l'algorithme sont réparties en blocs de taille p, où p représente le nombre de processeurs disponibles. Dans chaque bloc, les mêmes p propositions sont fournies à la chaîne dans p ordres différents. De plus des techniques de Rao–Blackwellisation sont utilisées sur la longueur du bloc pour réduire le bruit venant des tirages uniformes. Il en résulte un coût additionnel négligeable par rapport à l'algorithme initial, qui ne dépend pas du nombre total d'itérations de l'algorithme mais seulement de la taille p des blocs, à l'inverse d'autres techniques de Rao–Blackwellisation existantes dont le coût croît avec le nombre d'itérations.

La méthode, appelée Metropolis–Hastings indépendant par bloc, est décrite puis illustrée sur un exemple jouet et un exemple de régression probit, où les gains obtenus sont d'autant plus importants que le taux d'acceptation de l'algorithme de Metropolis–Hastings à proposition indépendante est bas, ce qui est souvent le cas en pratique.

Authors
- Pierre E. Jacob (Université Paris-Dauphine, CREST, Paris),
- Christian P. Robert (Université Paris-Dauphine, CREST, Paris),
- Murray H. Smith (NIWA, Wellington)

Status Published in the *Journal of Computational and Graphical Statistics*, Volume 20, Issue 3, 2011, pages 616-635.

Abstract

In this paper, we consider the implications of the fact that parallel raw-power can be exploited by a generic Metropolis–Hastings algorithm if the proposed values are independent from the current value of the Markov chain. In particular, we present improvements to the independent Metropolis–Hastings algorithm that significantly decrease the variance of any estimator derived from the MCMC output, at a null computing cost since those improvements are based on a fixed number of target density evaluations that can be produced in parallel. The techniques developed in this paper do not jeopardize the Markovian convergence properties of the algorithm, since they are based on the Rao–Blackwell principles of [GS90], already exploited in [CR96], [AP05] and [DR10]. We illustrate those improvements both on a toy normal example and on a classical probit regression model, but stress the fact that they are applicable in any case where the independent Metropolis–Hastings is applicable.

Keywords: Markov Chain Monte Carlo, independent Metropolis–Hastings, parallel computation, Rao-Blackwellization, permutation.

2.1 Introduction

The Metropolis–Hastings (MH) algorithm provides an iterative and converging scheme to sample from a complex target density π. Each iteration of the algorithm generates a new value of the Markov chain that relies on the result of the previous iteration. The underlying Markov principle is well-understood and leads to a generic convergence principle as described, e.g., in [RC04]. However, due to its Markovian nature, this algorithm is not straightforward to parallelize, which creates difficulties in slower languages like R [R D06]. Nevertheless, the increasing number of parallel cores that are available at a very low cost drives more and more interest in "parallel-friendly" algorithms, that is, in algorithms that can benefit from the available parallel processing units on standard computers (see. e.g., [HDL$^+$11], [LYG$^+$09], [SWC$^+$10]).

Different techniques have already been used to enhance some degree of parallelism in generic Metropolis–Hastings (MH) algorithms, beside the basic scheme of running p MCMC algorithms independently in parallel and merging the results. For instance, a natural entry is to rely on renewal properties of the Markov chain [MTY95, Rob95, HJPR02], waiting for all p chains to exhibit a renewal event and then using the blocks as iid, but the constraint of Markovianity cannot be removed. [Ros00] also points out the difficult issue of accounting for the burn-in time: while, for a single MCMC run, the burn-in time is essentially negligible, it does create a significant bias when running parallel chains (unless perfect sampling can be implemented). [CM05] mix antithetic coupling and stratification with perfect sampling. Using a different approach, [CRY09] rely on p parallel chains to build an adaptive MCMC algorithm, considering in essence that the product of the target densities over the chains is their target, a perspective that obviously impacts the convergence properties of the multiple chain. [CGK06] take advantage of parallelization to build a non-reversible algorithm that can avoid the scaling effect of specific neighborhood structures, hence focussing on a very special type of problem.

A particular family of MH algorithm is the Independent Metropolis–Hastings (IMH) algorithm, where the proposal distribution (and hence the proposed value) does not depend on the current state of the Markov chain. Due to this characteristic, this specific algorithm is easier to parallelize and can therefore be considered as a good building block toward efficient parallel Markov Chain Monte Carlo algorithms, as will be explained in Section 2.2. We will focus on cases where the computation of the likelihood function constitutes the major part of the execution time in the MH algorithm. A most realistic example of this setting is provided in [WKB+09], where the model is based on a very complex Fortran program translating the results of several cosmological experiments, hence highly demanding in computing time. In this model, [WKB+09] use adaptive importance sampling and massive parallelization, rather than MCMC.

The fundamental idea in the current paper is that one can take advantage of the parallel abilities of arbitrary items of computing machinery, from cloud computing to graphical cards (GPU), in the case of the generic IMH algorithm, producing an output that corresponds to a much improved Monte Carlo approximation machine at the same computational cost. The techniques presented here are related with those explained in [Per99] and more closely to those in [AP05], Section 3.1, since these authors condition upon the order statistic of the values proposed by the IMH algorithm, although in those earlier papers the links with parallel computation were not established and hence the implementation of the Rao-Blackwellization scheme became problematic for long chains.

The plan of the paper is as follows: the standard IMH algorithm is recalled in Section 2.2, followed by a description of our improving scheme, called here “block Independent Metropolis–Hastings” (block IMH). This improvement depends on a choice of permutations on $\{1, \dots, p\}$ that is described in details in Section 2.3. We demonstrate the connections between block IMH and Rao–Blackwellization in Section 2.4. Results for a toy example are presented throughout the paper and a realistic probit regression example is described in Section 2.5 as an illustration of the method.

2.2 Improving the IMH algorithm

2.2.1 Standard IMH algorithm

We recall here the basic IMH algorithm, assuming the availability of a proposal distribution that we can sample, and which probability density μ is known up to a normalization constant. The independent Metropolis–Hastings algorithm, described in Algorithm 1, generates a Markov chain with invariant density π, corresponding to the target distribution.

Algorithm 1 IMH algorithm

```
Set $x_0$ to an arbitrary value
for $t = 1$ to $T$ do
    Generate $y_t \sim \mu$
    Compute the ratio:
        $\rho(x_{t-1}, y_t) = \min\left\{1, \frac{\pi(y_t)}{\mu(y_t)} \frac{\mu(x_{t-1})}{\pi(x_{t-1})}\right\}$
    Set $x_t = y_t$ with probability $\rho(x_{t-1}, y_t)$; otherwise set $x_t = x_{t-1}$
end for
```

In the larger picture of Monte Carlo and MCMC algorithms, the IMH algorithm holds a rather special status. It has certainly been studied more often than other MCMC schemes [RC04], but it is undoubtedly a less practical solution than the more generic random walk Metropolis–Hastings algorithm. For instance, it is rather rarely used by itself because it requires the derivation of a tolerably good approximation to the true target, approximation that most often is unavailable. On the other hand, first-order approximations and Metropolis-within-Gibbs schemes are not foreign to calling for IMH local moves based on Gaussian representations of the targets. The reason theoretical studies of the IMH algorithm abound is that it has strong links with the non-Markovian simulation methods such as importance sampling. Contrary to random-walk Metropolis–Hastings schemes, IMH algorithms may enjoy very good convergence properties and may also reach acceptance probabilities that are close to one. Furthermore, the potentially large gain in variance reduction provided by the parallelization scheme developped in this paper may counteract the lesser efficiency of the original IMH compared with the random walk Metropolis–Hastings algorithm.

An important feature of the IMH algorithm, when addressing parallelism, is that it cannot work but in an iterative manner, since the outcome of step t, namely the value x_t, is required to compute the acceptance ratio at step $t+1$. This sequential construction is compulsory for the validation of the algorithm given the Markov property at its core [RC04]. Nonetheless, given that, in the IMH algorithm, the proposed values (y_t) are generated independently from the current state of the Markov chain, x_t, it is altogether possible to envision the generation of T proposed values y_t first, along with the computation of the associated ratios $\omega_t = \pi(y_t)/\mu(y_t)$. Once this computation requirement is completed, only the acceptance steps need to be considered iteratively. This two-step perspective makes for a huge saving in computing time when the simulation of the y_t's and the derivation of the ω_t's can be achieved in parallel since both the remaining computation of the ratios $\rho(x_{t-1}, y_t)$ given the ω_t's and their subsequent comparison with uniform draws typically are orders of magnitude faster.

In this respect the IMH algorithm strongly differs from the random walk Metropolis–Hastings (RWMH) algorithm, for which acceptance ratios cannot be processed beforehand because the proposed simulated values depend on the current value of the Markov chain. The universal availability of parallel processing schemes may thus lead to a new surge of popularity for the IMH algorithm. Indeed, when taking advantage of p parallel processing units, an IMH can be run for p times as many iterations as RWMH, at almost the same computing cost since RWMH cannot be directly parallelized.

In order to better describe this increased computing power, we first note that, once T successive values of a Markov chain have been produced, the sequence is usually processed as a regular Monte Carlo sample to obtain an approximation of an expectation under the target distribution, $\mathbb{E}_\pi[h(X)]$ say, for some arbitrary functions h. We propose in this paper a technique that improves the precision of the estimation of this expectation by taking advantage of parallel processing units without jeopardizing the Markov property.

Before presenting our improvement scheme, we introduce the notation $\vee$ (read "or") for the operator that represents a single step of the IMH algorithm. Using this notation, given the current value x_t and a sequence of p independent proposed values $y_1, \ldots, y_p \sim \mu$, the IMH algorithm goes from step t to step $t+p$ according to the diagram in Figure 2.1.

$$\begin{tikzcd}
x_t \arrow[r] & x_{t+1} := x_t \vee y_1 \arrow[r] & x_{t+2} := x_{t+1} \vee y_2 \arrow[r] & \cdots \arrow[r] & x_{t+p} := x_{t+p-1} \vee y_p
\end{tikzcd}$$

Figure 2.1: IMH steps between iteration t and iteration $t+p$.

2.2.2 Block IMH algorithm

We propose to take full advantage of the simulated proposed values and of the computation of their corresponding ω ratios. To this effect, we introduce the *block IMH algorithm*, made of successive simulations of blocks of size $p \times p$. In this alternative scheme, the number of blocks b is such that the number of desired iterations T is equal to $b * p$, in order to keep the comparison with a standard IMH output fair. Usually p needs not be calibrated since it represents the number of physical parallel processing units that can be exploited by the code. However, in principle, this number p can be set arbitrarily high and based on virtual parallel processing units, the drawback being then an increase in the computing cost. (Note that the block IMH algorithm can also be implemented with no parallel abilities, still it provides a gain in variance that may counteract the increase in time.) In the following examples, we take p varying from 4 to 100. We first explain how a block is simulated, and then how to move from one block to the next.

A $p \times p$ block consists in the generation of p parallel generations of p values of Markov chains, all starting at time t from the current state x_t and all based on the *same* proposed simulated values $y_1, \ldots, y_p$. The different between the p flows is the orders in which those y_i's are included. For instance, these orders may be the p circular permutations of $y_1, \ldots, y_p$, or they may be instead random permutations, as discussed in detail (and compared) in Section 2.3. The block IMH algorithm is illustrated in Figure 2.2 for the circular set of permutations.

$$\begin{tikzcd}
& x_{t+1}^{(1)} := x_t \vee y_1 \arrow[r] & x_{t+2}^{(1)} := x_{t+1}^{(1)} \vee y_2 \arrow[r] & \cdots \arrow[r] & x_{t+p}^{(1)} := x_{t+p-1}^{(1)} \vee y_p \\
& x_{t+1}^{(2)} := x_t \vee y_2 \arrow[r] & x_{t+2}^{(2)} := x_{t+1}^{(2)} \vee y_3 \arrow[r] & \cdots \arrow[r] & x_{t+p}^{(2)} := x_{t+p-1}^{(2)} \vee y_1 \\
x_t \arrow[uur, bend left] \arrow[ur, bend left] \arrow[r] \arrow[dr, bend right] \arrow[ddr, bend right] & \cdots & \cdots & \cdots & \cdots \\
& x_{t+1}^{(p-1)} := x_t \vee y_{p-1} \arrow[r] & x_{t+2}^{(p-1)} := x_{t+1}^{(p-1)} \vee y_p \arrow[r] & \cdots \arrow[r] & x_{t+p}^{(p-1)} := x_{t+p-1}^{(p-1)} \vee y_{p-2} \\
& x_{t+1}^{(p)} := x_t \vee y_p \arrow[r] & x_{t+2}^{(p)} := x_{t+1}^{(p)} \vee y_1 \arrow[r] & \cdots \arrow[r] & x_{t+p}^{(p)} := x_{t+p-1}^{(p)} \vee y_{p-1}
\end{tikzcd}$$

$p \times p$ block

Figure 2.2: Block simulation from step $t+1$ to step $t+p$. Here, circular permutations of the proposed values are used for illustration purposes.

It should be clear that each of the p parallel chains found in this block is a valid MCMC sequence of length p when taken separately. As such, it can be processed as a regular MCMC output. In particular, if x_t is simulated from the stationary distribution, any of the subsequent $x_{t+i}^{(j)}$ is also simulated from the stationary distribution. However, the point of the p parallel flows is double:

- it aims at integrating out part of the randomness resulting from the ancillary order in which the y_k's are chosen, getting close to the conditioning on the order statistics of the y_k's advocated by [Per99];
- it also aims at partly integrating out the randomness resulting from the generation of uniform variables in the selection process, since the block implementation results in drawing p^2 uniform realizations instead of p uniform realizations for a standard IMH setting.

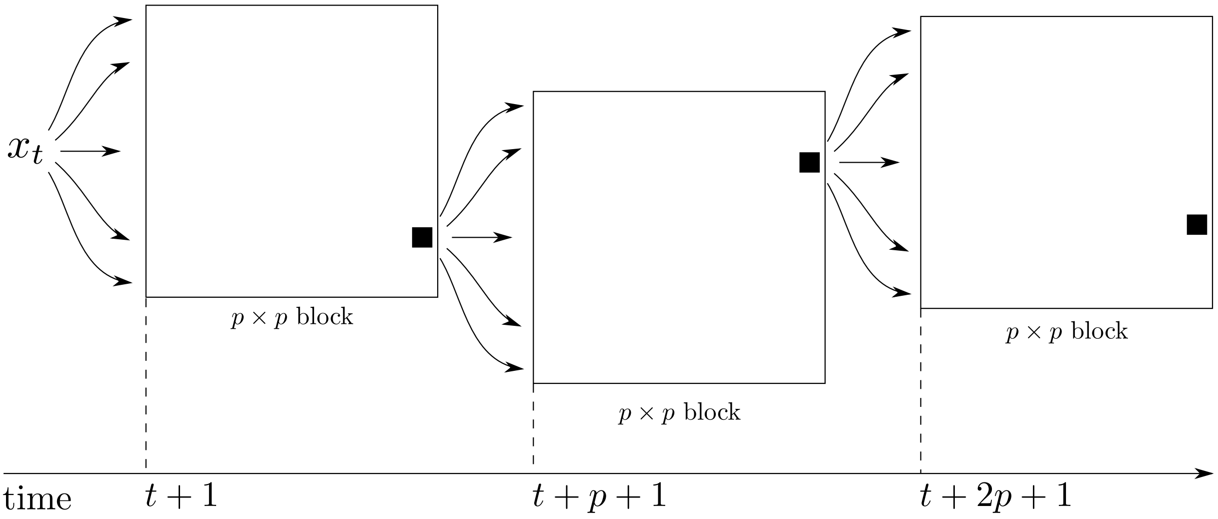


Figure 2.3: The block IMH algorithm runs p parallel chains during p steps, then picks one of the final values (represented by the black squares) and iterates. (An alternative transition mechanism involves sampling randomly one of the p^2 terms within the block.)

Both of those points essentially amount to implementing a new Rao--Blackwellization technique (a more precise connection is drawn in Section 2.4). In an independent setting, each of the y_k's occurs a number $n_k \geq 0$ of times across the p steps of the p parallel chains, i.e. for a number p^2 of realizations. Therefore, when considering the standard estimator $\hat{\tau}_1$ of $\mathbb{E}_\pi\left[h(X)\right]$, based on a *single* MCMC chain,

$$\hat{\tau}_1(x_t, y_{1:p}) = \frac{1}{p}\sum_{k=1}^{p} h(x_{t+k})$$

this estimator necessarily has a larger variance than the double average

$$\hat{\tau}_2(x_t, y_{1:p}) = \frac{1}{p^2}\sum_{j=1}^{p}\sum_{k=1}^{p} h(x_{t+k}^{(j)}) = \frac{1}{p^2}\sum_{k=0}^{p} n_k h(y_k)$$

where $y_0 := x_t$ and n_0 is the number of times x_t is repeated. (The proof for the reduction of the variance from $\hat{\tau}_1$ to $\hat{\tau}_2$ easily follows from a double integration argument.) We again insist on the compelling feature that computing $\hat{\tau}_2$ using p parallel processing units does not cost more time than computing $\hat{\tau}_1$ using a single processing unit.

In order to preserve its Markov validation, the algorithm must properly continue at time $t+p$. An obvious choice is to pick one of the p sequences at random and to take the corresponding $x_{t+p}^{(j)}$ as the value of x_{t+p}, starting point of the next parallel block. This mechanism is represented in Figure 2.3. While valid from a Markovian perspective, since the sequences are marginally produced by a regular IMH algorithm, this means that the chain deduced from the block IMH algorithm is converging at *exactly* the same speed as the original IMH algorithm. An alternative choice for the starting points of the blocks takes advantage of the weights n_k on the y_k's that are computed via the block structure. Indeed, those weights essentially act as importance weights and they allow for a selection of any of the p^2 $x_{t+i}^{(j)}$'s as the starting point of the incoming block, which corresponds to choosing one of the proposed y_k's with probability proportional to n_k. While this proposal does reduce the length of the resulting chain, it does not impact the estimation aspects (which still involve all of the p^2 values) and it could improve convergence, given that the weighted y_k's behave like a discretized version of a sample from the target density π. We will not cover this alternative any further.

The original version of the block IMH algorithm is described in Algorithm 2, The algorithm is made of a loop on the b blocks and an inner loop on the p parallel chains of each block. The p steps of this inner loop are actually meant to be implemented in parallel. The output of Algorithm 2 is double:

- a standard Markov chain of length T, which is made of b chains of length p, each of which is chosen among p chains at line 12 of Algorithm 2,
- a $p \times T$ array $(x_t^k)_{t=1:T}^{k=1:p}$, on which the estimator $\hat{\tau}_2$ is based.

Algorithm 2 block IMH algorithm

```
 1: Set $x_0$ to an arbitrary value, compute $\omega_0$
 2: Set $x_{\text{start}} = x_0$, $\omega_{\text{start}} = \omega_0$
 3: Set a block size $p$, and a number of blocks $b$, such that $b * p = T$
 4: Generate all proposed values $y_1, \ldots, y_T \sim \mu$
 5: Compute all ratios $\omega_1, \ldots, \omega_T$
 6: for $i = 1$ to $b$ do
 7:     Choose $p$ permutations $\sigma_1, \ldots, \sigma_p$
 8:     for $k = 1$ to $p$ do
 9:         Run $p$ steps of an IMH given:
              • $(x_{\text{start}}, \omega_{\text{start}})$
              • $p$ proposed values $y_{(i-1)*p+1}, \ldots, y_{i*p}$ shuffled with the permutation $\sigma_k$
              • the $p$ corresponding ratios $\omega_j$'s
10:         Save as $x^{(k)}_{(i-1)*p+1}, \ldots, x^{(k)}_{i*p}$ the resulting chain
11:     end for
12:     Draw an index $j$ uniformly in $\{1, \ldots, p\}$, set $x_{\text{start}} = x^{(j)}_{i*p}$, set $\omega_{\text{start}}$ as the corre-
        sponding ratio $\omega$.
13: end for
```

2.2.3 Savings on computing time

Since the point-wise evaluation of the target density $\pi(y_k)$ is usually the most computer-intensive part of the algorithm, sampling additional uniform variables has a negligible impact here, as do further costs related to the storage of vectors larger than in the original IMH. This is particularly compelling since the multiple chains do not need to be stored further than during a single block execution time. That is why, although we sample p times more uniforms in the block IMH algorithm, we still consider it to be running at roughly of the same cost as the IMH algorithm. The number of target density evaluations indeed is the same for both and most often represent the overwhelming part of the computing time in the Metropolis–Hastings algorithm. Besides, pseudo-random generation of uniforms can also benefit from parallel processing, see e.g. [LSCK01].

In the following Monte Carlo experiment, various versions of the block IMH algorithm are compared one to another, as well as to standard IMH and importance sampling. We stress that a straightforward reason for not conducting a comparison with a plain parallel algorithm based on p independent parallel chains is that it does not make much sense cost-wise. Indeed, running p parallel MCMC chains of the same length T does cost p

times more in terms of target density evaluations. Obviously, if one insists on running p independent chains, for instance as to initialize an MCMC algorithm from several well-dispersed starting points, each of those chains can benefit from our stabilizing method, which will improve the resulting estimation.

The method is presented here for square blocks of dimension (p, p), but blocks could be rectangular as well: the algorithm is equally valid when using $r \neq p$ permutations, leading to (r, p) blocks. We focus here on square blocks because when the machine at hand provides p parallel processing units, then it is most efficient to simulate the proposed values and the uniforms, and to compute the target densities and the acceptance ratios at the p proposed values in parallel. Once again, the block IMH algorithm with $p \times p$ square blocks has about the same cost as the original IMH algorithm, because computing target densities and acceptance ratios does more than compensate for the cost of randomly picking an index at the end of each block. This amounts to say that line 4 of Algorithm 1 and line 5 of Algorithm 2 are (by far) the most computationally demanding ones in the respective algorithms.

2.2.4 Toy example

We now introduce a toy example that we will follow throughout the paper. The target π is the density of the standard $\mathcal{N}(0, 1)$ normal distribution and the proposal μ is the density of the $\mathcal{C}(0, 1)$ Cauchy distribution. Hence, the density ratio is

$$\omega(x) = \frac{\pi(x)}{\mu(x)} \propto (1 + x^2) \exp\left(-\frac{1}{2}x^2\right)$$

We only consider the integral $\int x\pi(\mathrm{d}x)$, the mean of π equal to zero in this case. The acceptance rate of the IMH algorithm for this example is around 70%. (Note that IMH with higher acceptance rates are considered to be more efficient, in opposition to other Metropolis–Hastings algorithms, see RC04.)

In all results related to the toy example presented thereafter, $10,000$ independent runs are used to compute the variance of the estimates. The value of p represents the number of parallel processing units that are available, ranging from 4 for a desktop computer to 100 for a cluster or a graphics processing unit (GPU) (this value could even be larger for computers equipped with multiple GPUs).

The results of the simulation experiments are presented in Figures 2.4–2.8 as barplots, which indicate the percentage of variance decrease associated with the estimators under comparison, the reference estimator being the standard IMH output $\hat{\tau}_1$. In agreement with the block sampling perspective, the same proposed values and uniform draws were used for all the estimators that are plotted on the same graph (that is, for a given value of p), so that the comparison is not perturbed by an additional noise associated with the simulation.

2.3 Permutations

While the choice of permutations in line 7 of Algorithm 2 is irrelevant for the validation of the parallelization, it has important consequences on the variance improvement and we now discuss several natural choices. The idea of testing various orders of the proposed values in a IMH algorithm appeared in [AP05] where the permutations were chosen to be circular. We first list natural types of permutations along with some justifications, and then we empirically compare their impact on estimation performances for the toy example.

2.3.1 Five natural permutations

Let $\mathcal{S}$ be the set of permutations of $\{1, \dots, p\}$. Its size is $p!$, therefore too large to allow for averaging over all permutations, although this solution would be ideal. We consider the simpler option of finding p efficient permutations in $\mathcal{S}$, denoted by $(\sigma_1, \dots, \sigma_p)$, the goal being a choice favoring the largest possible decrease in the variance of the estimator $\hat{\tau}_2$ defined in Section 2.2.

Same order

The most basic choice is to pick the same permutation on each of the p chains:

$$\sigma_1 = \sigma_2 = \dots = \sigma_p$$

This selection may sound counterproductive, but we still obtain a significant decrease in the variance of $\hat{\tau}_2$ using this set of permutations, when compared with $\hat{\tau}_1$. The reason for the improvement is that p times more uniforms are used in $\hat{\tau}_2$ than in $\hat{\tau}_1$, leading to a natural Rao-Blackwellization phenomenon that is studied in details in Section 2.4. Nonetheless this simplistic set of permutations is certainly not the best choice since it does not integrate out the ancillary randomness resulting from the arbitrary ordering of the proposed values.

Circular permutations

Another simple choice is to use circular permutations. For $1 \leq i \leq p$, we define

$$\sigma_i(1) = i, \sigma_i(2) = i + 1, \dots, \sigma_i(p - i + 1) = p, \sigma_i(p - i + 2) = 1, \dots, \sigma_i(p) = i - 1.$$

An appealing property of the circular permutations is that each simulated value y_k is proposed and evaluated at a different step for each chain. However, a drawback is that the order is not deeply changed: for instance y_{k-1} will always be proposed one step before y_k except for one of the p chains, for which y_k is proposed first.

Random permutations

A third choice is to use random orders, that is random shufflings of the sequence $\{1, \dots, p\}$. We can either draw those random permutations with or without replacement in the set $\mathcal{S}$, but considering the cardinality of the set $\mathcal{S}$ this does not make a large difference. Indeed, it is unlikely to draw twice the same permutation, except for very small values of p.

Half-random half-reversed permutations

A slightly different choice of permutations consists in drawing $p/2$ permutations at random (p is taken to be even here to simplify the notations). Then, denoting the first $p/2$ permutations by $\sigma_1, \dots, \sigma_{p/2}$, we define for $1 \leq k \leq p/2$:

$$\sigma_{k+p/2}(1) = \sigma_k(p), \sigma_{k+p/2}(2) = \sigma_k(p - 1), \dots \sigma_{k+p/2}(p) = \sigma_k(1)\,.$$

The motivation for this inversion of the orders is that, in the second half of the permutations, the opposition with the "reversed" first half is maximal. This choice, suggestion of Art Owen (personal communication), aims at minimizing the possible common history among the p parallel chains. Indeed two chains with the same proposed values in reverse order cannot have a common path of length more than 1.

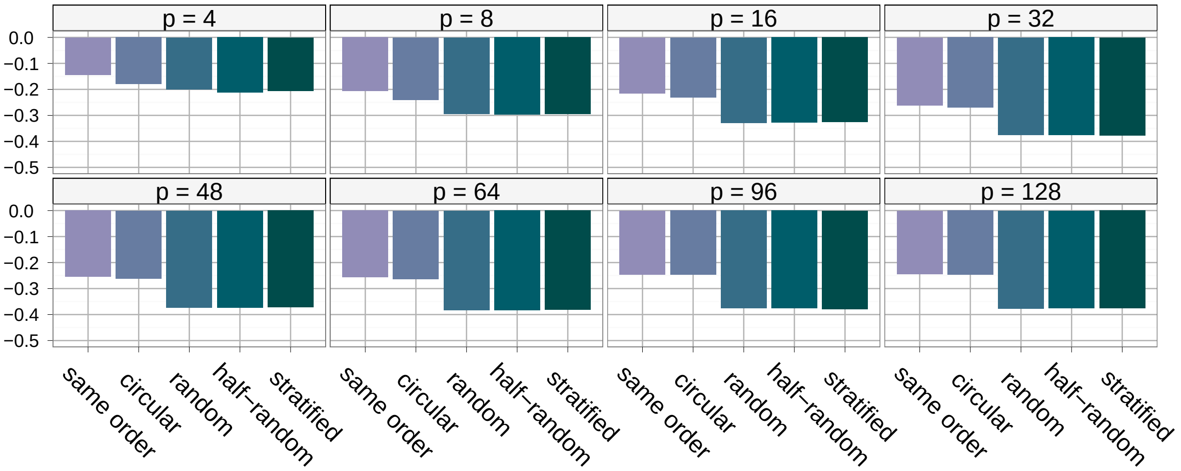


Figure 2.4: Variance reductions, when compared with the basic estimator $\hat{\tau}_1$, of the various block estimators $\hat{\tau}_2$ associated with each permutation scheme for several values of p.

Stratified random permutations

Finally we can try to draw permutations that are far from one another in the set $\mathcal{S}$. For instance we can define the lexicographic order on $\mathcal{S}$, draw indices from a low discrepancy sequence on the set $\{1, \ldots, p!\}$ and select the permutations corresponding to these indices. In a simpler manner, we do use here a stratified sampling scheme: we first draw a random permutation conditionally on its first element being 1, then another permutation beginning with 2, and so forth until the last permutation which begins with p.

2.3.2 A Monte Carlo comparison

We compare the five described types of permutations on the toy example. Figure 2.4 shows the results for various values of p, displaying the variance reduction of $\hat{\tau}_2$ associated with each of the permutation orders, compared to the variance of the original IMH estimator $\hat{\tau}_1$. For each of the $10,000$ independent replications, the block IMH algorithm was launched on one single $p \times p$ block, e.g. with $b = 1$ using the notation of Section 2.2, since b plays no role whatsoever in this comparison.

As mentioned above, using the same order in the y_k's for each of the p parallel chains already produces a significant decrease of about 20% in the variance of the estimators. This simulation experiment shows that the three random permutations (random, half-random half-reversed and stratified) are quite equivalent in terms of variance improvement and that they are significantly better than the circular permutation proposal, which only slightly improves over the “same order” scheme. Therefore, in the next Monte Carlo experiments, we will only use the random order solution, simplest of the random schemes. An amount of improvement like 35% when $p \geq 32$ is quite impressive when considering that it is essentially obtained cost-free for a computer with parallel abilities [HDL+11].

2.4 Rao–Blackwellization

Another generic improvement that can be brought over classical MH algorithms is Rao–Blackwellization [GS90, CR96]. In this section, two Rao–Blackwellization methods are presented, one that is computationally free and one that, on the contrary, is computationally expensive. We then implement both solutions within the block IMH algorithm and explain why the "same order" scheme already improves upon the IMH algorithm.

2.4.1 Primary Rao–Blackwellization

Within the standard IMH algorithm of Section 2.2.1, a cost-free improvement can be obtained by a straightforward Rao–Blackwellization argument. Given that at step $t+i$, y_i is accepted with probability $\rho(x_{t+i-1}, y_i)$ and rejected with probability $1-\rho(x_{t+i-1}, y_i)$, the weight of y_i can be updated by $\rho(x_{t+i-1}, y_i)$ and the weight of the simulated value y_j corresponding to x_{t+i-1} can be similarly updated by the probability $1-\rho(x_{t+i-1}, y_i)$. Considering next the block IMH algorithm, at the beginning of each block we can define p weights, denoted by $(w_k)_{k=1}^p$, initialized at 0 and then, for the first of the p parallel chains, denoting by j the index such that $x_{t+i-1}^{(1)} = y_j$, we update these weights at each time $t+i$ as

$$\begin{aligned} w_j &\leftarrow w_j + 1 - \rho(x_{t+i-1}^{(1)}, y_i) \\ w_i &\leftarrow w_i + \rho(x_{t+i-1}^{(1)}, y_i) \end{aligned}$$

This is obviously repeated for each of the other parallel chains, ending up with $\sum_k w_k = p^2$. This leads to a new estimator

$$\hat{\tau}_3(x_t, y_{1:p}) = \frac{1}{p^2} \sum_{k=0}^{p} w_k h(y_k)\,.$$

This estimator still depends on all uniform generations created within the block, since those weights w_k depend upon the acceptances and rejections of the y_k's made during the block update. However, along the steps of the block, the w_k's are basically updated by the expectations of the acceptance indicators conditionally upon the results of the previous iterations, whereas the n_k of Section 2.2 are directly updated according to the acceptance indicators. Hence, the w_k's have a smaller variance than the n_k's by virtue of the Rao–Blackwell theorem, leading to $\hat{\tau}_3$ necessarily having a smaller variance than $\hat{\tau}_2$.

We now discuss a more involved Rao-Blackwellization technique first proposed by [CR96].

2.4.2 Block Rao–Blackwellization

Exploiting the Rao–Blackwellization technique of [CR96] within each parallel chain provides via a conditioning argument an even more stable approximation of arbitrary posterior quantities. As developed in [CR96], for a single Markov chain $(x_1^{(i)}, \ldots, x_p^{(i)})$, a Rao–Blackwell weighting scheme on the proposed values y_t, with weights φ_t, is given by a recursive scheme

$$\varphi_t^{(i)} = \delta_t \sum_{j=t}^{p} \xi_{tj}$$

where $(t > 0)$

$$\delta_0 = 1\,, \qquad \xi_{tt} = 1\,, \qquad \xi_{tj} = \prod_{u=t+1}^{j} (1-\rho_{tu})$$

and

$$\delta_t = \sum_{j=0}^{t-1} \delta_j \xi_{j(t-1)} \rho_{jt} \,,$$

associated with the Metropolis–Hastings ratios

$$\omega_t = \pi(y_t)/\mu(y_t)\,, \qquad \rho_{tu} = \omega_u/\omega_t \wedge 1\,.$$

The cumulated computation of the δ_t's, of the ρ_{tu}'s and of the ξ_{tu}'s requires an $\mathrm{O}(p^2)$ computing time. Given that p is usually not very large, this additional cost is not redhibitory as in the original proposal of [CR96] who were considering the application of this Rao–Blackwellization technique over the whole chain, with a cost of $\mathrm{O}(T^2)$ (see also Per99).

Therefore, starting from the estimator $\hat{\tau}_2$, the weight n_k counting the number of occurrences of y_k in the $p \times p$ block can be replaced with the expected number φ_k of times y_k occurs in this block (given the p proposed values), which is the sum of the expected numbers of times y_k occurs in each of the p parallel chain:

$$\varphi_k = \sum_{i=1}^{p} \varphi_k^{(i)}$$

Since the p parallel chains incorporate the proposed values with different orders, the φ's may differ for each chain and must therefore be computed p times. Note that the cost is still in $\mathrm{O}(p^2)$ if this computation can be implemented in parallel. Then, by a Rao-Blackwell argument, $\hat{\tau}_2$ and τ_3 are dominated by $\hat{\tau}_4$ defined as follows:

$$\hat{\tau}_4(x_t, y_{1:p}) = \frac{1}{p^2} \sum_{k=0}^{p} \varphi_k h(y_k)$$

Therefore, this Rao–Blackwellization scheme involves *no* uniform generation for the computation of $\hat{\tau}_4$: the randomness associated with these uniforms is completely integrated out.

The four estimators defined up to now can be summarized as follows:

- $\hat{\tau}_1$ is the basic IMH estimator of $\mathbb{E}_\pi\left[h(X)\right]$,
- $\hat{\tau}_2$ improves $\hat{\tau}_1$ by averaging over permutations of the proposed values, and by using p times more uniforms than $\hat{\tau}_1$,
- $\hat{\tau}_3$ improves upon $\hat{\tau}_2$ by a basic Rao-Blackwell argument,
- $\hat{\tau}_4$ improves upon the above by a further Rao-Blackwell argument, integrating out the ancillary uniform variables, but at a cost of $\mathrm{O}(p^2)$.

Note that these four estimators all involve the same number p of target density evaluations, which again represent the overwhelming part of the computing time.

2.4.3 A numerical evaluation

Figure 2.5 gives a comparison between the variances of the three improved estimators defined above and the variance of the basic IMH estimator. The permutations are random in this case. As was already apparent on Figure 2.4, the block estimator $\hat{\tau}_2$ is significantly better than $\hat{\tau}_1$ for any value of p. Moreover, both Rao-Blackwellization modifications seem

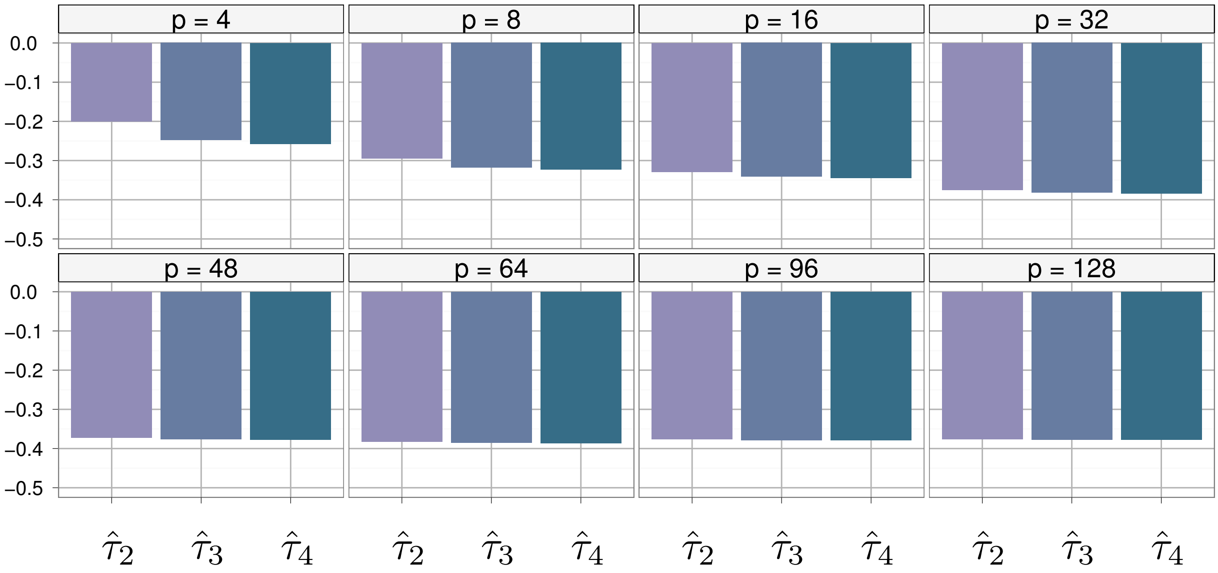


Figure 2.5: Variance improvement over the basic estimator $\hat{\tau}_1$ for three improved block IMH estimators.

to improve only very slightly the estimation when compared with $\hat{\tau}_2$, even though the improvement increases with p.

Recall that the “same order” scheme already provided a significant decrease in the variance of the estimation. In the light of our results, our interpretation is that using p parallel chains with the same proposed values acts like a ”poor man” Rao–Blackwellization technique since p times more uniforms are used. Specifically, each of the p proposed values is proposed p times instead of once, thus reducing the impact of each single uniform draw on the overall estimation.

When we use Rao–Blackwellization on top of the block IMH, in the estimators $\hat{\tau}_3$ and $\hat{\tau}_4$, we try indeed to integrate out a randomness that already is partly gone. This explains why, although Rao–Blackwellization techniques provide a significant improvement over standard IMH, the improvement is much lower and thus rather unappealing when used in the block IMH setting. This outcome was at first frustrating since Rao–Blackwellization is indeed affordable at a cost of only $O(p^2)$. However, this shows *in fine* that the improvement brought by the block IMH algorithm roughly provides the same improvement as the Rao–Blackwell solution, at a much lower cost.

2.4.4 Comparison with Importance Sampling

The proposal density μ may also be used to construct directly an importance sampling (IS) estimator

$$\hat{\tau}_{IS} = \frac{1}{T}\sum_{t=1}^{T} h(y_t)\frac{\pi(y_t)}{\mu(y_t)} ,$$

where the values y_t are drawn from μ. It therefore makes sense to compare the IMH algorithm with an IS approximation because IS is similarly easy to parallelize, and straightforward to program. Furthermore, since the IS estimator does not involve ancillary uniform variables, it is comparable to the Rao–Blackwellized version of IMH, and hence to the block IMH. Obviously, IS cannot necessarily be used in the settings when IMH algorithms are used, because the latter are also considered for approximating simulations from the target density π. In particular, when considering Metropolis-within-Gibbs algorithms, IS cannot be used in a straightforward manner, even for approximating

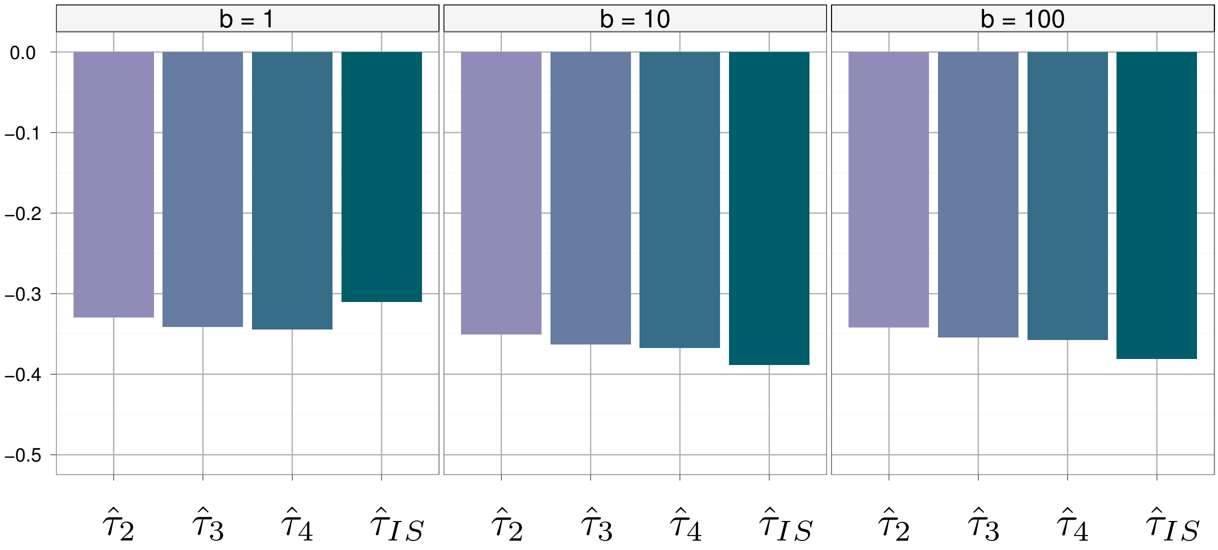


Figure 2.6: Variance reduction, when compared with the basic estimator $\hat{\tau}_1$, of the three improved block IMH estimators, and the IS estimator $\hat{\tau}_{IS}$, for $p = 16$ and $b = 1, 10, 100$.

integrals.

Before giving numerical results for a comparison run on the toy example, we now explain why in this comparison we took the number of blocks to be larger than 1. The previous comparisons were computed with $b = 1$, i.e. on a single $p \times p$ block and for a large number of independent runs. The choice of b was then irrelevant since we were comparing methods that were exploiting in different ways the p proposed values generated in each block. When considering the block IMH algorithm as a whole, as explained in Section 2.2, the end of each block sees a new starting value chosen from the current block. This ensures that the algorithm produces a valid Markov chain. However, our construction also implies that the successive blocks produced by the algorithm are correlated, which should lead to lesser performances than for the IS estimator.

In the comparison between IMH and IS, we therefore need to take into account this correlation between successive blocks. To this effect, we produce the variance reductions for several values of b. Those reductions are presented in Figure 2.6 for $p = 16$ and different values of $b = 1, 10, 100$. Once again, the permutations in the block IMH algorithm are chosen to be random.

Figure 2.6 shows the a priori surprising result that, when selecting $b = 1$ in the experiment, the variance results are in favor of the block IMH solutions over the IS estimator, but, for any realistic application, b is (much) larger than 1. For all $b \geq 10$, the IS estimator has a smaller variance than the three alternative block IMH estimators, if only by a small margin. (Note that the variance improvement over the original MCMC estimator is slightly increasing with b despite the correlation between blocks, given that the correlation between the p^2 terms involved in the block IMH estimators is lower than the correlation in the original MCMC chain.) This experiment thus shows that the block IMH solution gets very close to the IS estimator, while preserving the Markovian features of the original IMH algorithm.

2.5 A probit regression illustration

In order to evaluate the performances of the parallel processing presented in this paper on a realistic example, we examine its implementation for the probit model already analyzed

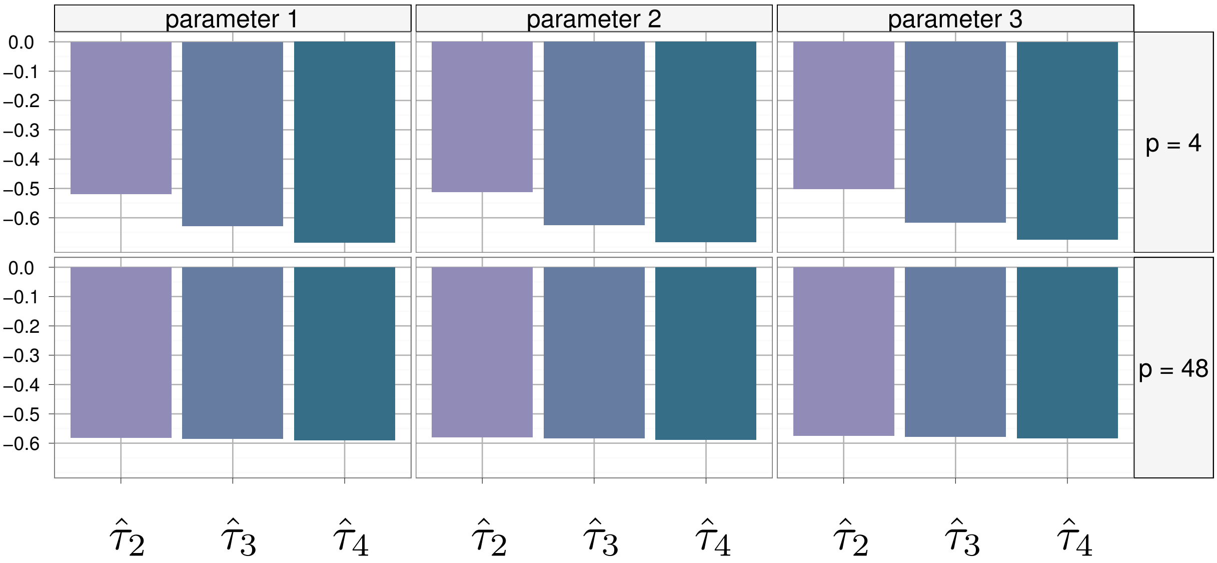


Figure 2.7: Variance reduction, when compared with the basic estimator $\hat{\tau}_1$, of the three improved block IMH estimators for $p = 10$ and each of the parameters.

in [MR10] for the comparison of model choice techniques because the “plug-in” normal distribution based on MLE estimates of the first two moments works perfectly as an independent proposal.

A probit model can be represented as a natural latent variable model in that, if we consider a sample $z_1, \ldots, z_n$ of n independent latent variables associated with a standard regression model, i.e. such that $z_i|\theta \sim \mathcal{N}\left(x_i^{\mathrm{T}}\theta, 1\right)$, where the x_i's are p-dimensional covariates and θ is the vector of regression coefficients, then $y_1, \ldots, y_n$ such that

$$y_i = \mathbb{I}_{z_i > 0}$$

is a probit sample. Indeed, given θ, the y_i's are independent Bernoulli rv's with $\mathbb{P}(y_i = 1|\theta) = \Phi\left(x_i^{\mathrm{T}}\theta\right)$ where Φ is the standard normal cdf. The choice of a prior distribution for the probit model is open to debate, but the above connection with the latent regression model induced [MR07] to suggest a g-prior model, $\theta \sim \mathcal{N}\left(0_p, n(X^{\mathrm{T}}X)^{-1}\right)$, with n as the g factor and X as the regressor matrix.

While a Gibbs sampler taking advantage of the latent variable structure is implemented in [MR10] and earlier references [AC93], a straightforward Metropolis--Hastings algorithm may be used as well, based on an independent proposal $\mathcal{N}(\hat{\theta}, c\widehat{\Sigma})$, where $\hat{\theta}$ is the MLE estimator, $\widehat{\Sigma}$ its asymptotic variance, and c a scaling factor.

As in [MR10] and [GC10], we use the R Pima Indian benchmark dataset, which contains medical information about 332 Pima Indian women with seven covariates and one explained binary diabetes variable [R D06].

For the purpose of illustrating the implementation of the block IMH algorithm, we only consider here three covariates, namely plasma glucose concentration (with coefficient θ_1), diastolic blood pressure (with coefficient θ_2) and diabetes pedigree function (with coefficient θ_3). We are interested in the posterior mean of those three regression parameters. In our experiment, we ran $10,000$ independent replications of our algorithm to produce a reliable evaluation of the variance of the estimators under comparison. In Figure 2.7 we present the variance comparison of the four estimators described in Section 2.4, for $p = 4$ and $p = 48$ and for each of the three regression parameters. In the independent proposal, the scale factor is chosen to be 3 since pilot runs showed that it led to an acceptance rate around 37%, with thus enough rejections to exhibit improvement by Rao–Blackwellization.

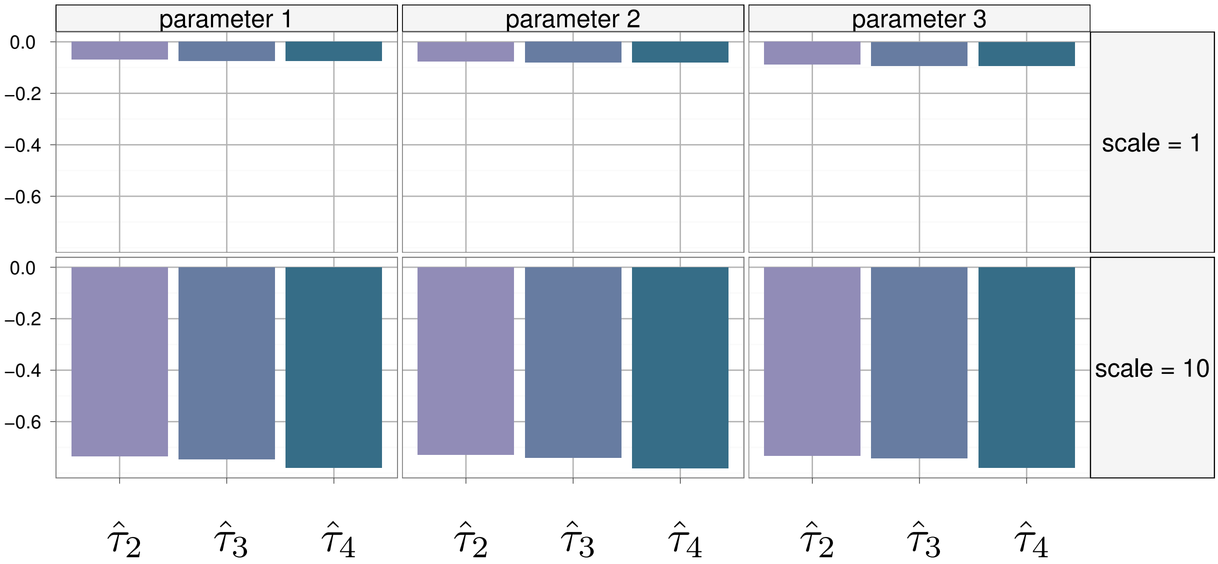


Figure 2.8: Variance comparison for $p = 16$ and two scaling factors: $c = 1$, with an associated acceptance rate of 96% (top) and $c = 10$, with an associated acceptance rate of 8% (bottom).

The results shown in Figure 2.7 confirm the huge decrease in variance previously observed in the toy example. The gains represented in those figures indicate that the block estimator $\hat{\tau}_2$ is nearly as good (in terms of variance improvement) as the Rao–Blackwellized block estimators $\hat{\tau}_3$ and $\hat{\tau}_4$, especially when p moves from 4 to 48. This confirms the previous interpretation given in Section 2.4 that the block IMH algorithm provides a cost-free Rao–Blackwellization as well as a partial averaging over the order of the proposed values.

The fact that the toy example showed decreases in the variance that were around 35% whereas the probit regression shows decreases around 60% is worth discussing. The amount of decrease in the variance differs from one example to the other, but it is more importantly depending on the acceptance rate of the Metropolis–Hastings algorithm. In fact, Rao–Blackwellization and permutations of the proposed values are useless steps if the acceptance rate is exactly 1. On the opposite, it may result in a significant improvement when the acceptance rate is low (since the part of the variance due to the uniform draws would then be much more important).

To illustrate the connection between the observed improvement and the acceptance rate, we propose in Figure 2.8 a variance comparison for $p = 16$ and two scaling factors c of the proposal covariance matrix in the probit regression model. In the previous experiment, we have used $c = 3$, which leads to an acceptance ratio around 37%. Here, if we take $c = 1$, the acceptance ratio rises to 96%, and hence almost all the proposed values are accepted. In this case permuting the proposed values and using Rao–Blackwellization techniques does not bring much of a variance decrease (Figure 2.8, top). On the other hand, if we take $c = 10$, the acceptance ratio drops down to 8% and the observed decrease in variance is huge. In this second case using all the proposed values gives much better results than relying on the standard IMH estimator, which is only based on 8% of the proposed values that were accepted (Figure 2.8, bottom).

The difference observed with this range of scaling factors is thus in agreement with the larger decrease in variance observed for the probit regression. This is a positive feature of the block IMH method, since in a complex model, the target distribution is most often poorly approximated by the proposal and thus the acceptance rate of the IMH algorithm is quite likely to be low.

2.6 Conclusion

The Monte Carlo experiments produced in this paper have shown that the proposed method improves significantly the precision of the estimation, when compared with the standard IMH algorithm. Beyond these examples, we see multiple situations where the block IMH algorithm can be used to improve the estimation in challenging problems. First, as already stated, the IMH algorithm can be used in Metropolis-within-Gibbs algorithms [GBT95]. Obviously if a single IMH step is performed for each component of the state, then the block IMH technique cannot be applied without incurring additional costs. However, it is also correct to update each component multiple times instead of once. Furthermore, a uniform Gibbs scan is rarely the optimal way to update the components and more sophisticated schemes have been studied, resulting in random scan Gibbs samplers and adaptive Gibbs samplers [LR10], where the probability of updating a given component depends on the component and is learned along the iterations of the algorithm. Hence if a component is updated n times more often than another, n IMH can be performed in a row, which allows the use of the block IMH technique with $p = n$.

IMH steps are also used within sequential Monte Carlo (SMC) samplers (Cho02, DD03), to diversify the particles after resampling steps. In this context, an independent proposal can be designed by fitting a (usually Gaussian) distribution on the particles. If the move step is repeated multiple times in a row, for instance to ensure a satisfying particle diversification, then the block IMH algorithm can be used.

Related to SMC, another context where the variance reduction provided by block IMH might be valuable is the class of particle Markov Chain Monte Carlo methods [ADH10]. For these methods, a particle filter is computed at each iteration of the MH algorithm to estimate the target density, and hence it is paramount to make the most out of the expensive computations involved by those estimates. This is thus a natural framework for parallelization.

As a final message, the block IMH method is close to being 100% parallel (except for the random draw of an index at the end of each block). Since parallel computing is getting increasingly easy to use, the free improvement brought by $\hat{\tau}_3$ is available for all implementations of the IMH algorithm. Furthermore, even without considering parallel computing, since the method uses the most of each target density evaluation, it brings significant improvement when computing the target density is very costly. In such settings, the cost of drawing p^2 instead of p uniform variates is negligible and the block IMH algorithm thus runs in about the same time as the standard IMH algorithm. We note that the time required to complete a block in the algorithm is essentially the maximum of the p times required to calculate the density ratios w_i. Therefore, if these times widely vary, there could be a diminishing saving in computation time as p increases for both the standard IMH and the block IMH algorithms. Nonetheless, even in such extreme cases, using $\hat{\tau}_3$ in the block IMH algorithm would bring a significant variance improvement at essentially no additional cost.

Acknowledgements

The work of the second author (CPR) was partly supported by the Agence Nationale de la Recherche (ANR, 212, rue de Bercy 75012 Paris) through the 2009 project ANR-08-BLAN-0218 Big'MC and the 2009 project ANR-09-BLAN-01 EMILE. Pierre Jacob is supported by a PhD fellowship from the AXA Research Fund. Since this research was initiated during the Valencia 9 Bayesian Statistics conference, the paper is dedicated to José Miguel Bernardo for the organization of this series of unique meetings since 1979. Discussions of the second author with participants during a seminar in Stanford University in August 2010 were quite helpful, in particular the suggestion made by Art Owen to include the "half-inversed half-random" permutations. The authors are grateful to Julien Cornebise for helpful discussions, in particular those leading to the stratified strategy, and to François Perron for his advice on the permutations. Comments and suggestions from the editorial team of JCGS were most helpful in improving the presentation of our results.

Bibliography

[AC93] J.H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *J. American Statist. Assoc.*, 88:669–679, 1993.

[ADH10] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo (with discussion). *J. Royal Statist. Society Series B*, 72, 2010. (to appear).

[AP05] Y.F. Atchadé and F. Perron. Improving on the independent Metropolis–Hastings algorithm. *Statistica Sinica*, 15:3–18, 2005.

[CGK06] Jukka Corander, Mats Gyllenberg, and Timo Koski. Bayesian model learning based on a parallel MCMC strategy. *Statistics and Computing*, 16(4):355–362, 2006.

[Cho02] N. Chopin. A sequential particle filter method for static models. *Biometrika*, 89:539–552, 2002.

[CM05] Radu V. Craiu and Xiao-Li Meng. Multiprocess parallel antithetic coupling for backward and forward Markov chain Monte Carlo. *Ann. Statist.*, 33(2):661–697, 2005.

[CR96] G. Casella and C.P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[CRY09] R.V. Craiu, J. Rosenthal, and C. Yang. Learn from thy neighbour: Parallel-chain and regional adaptive MCMC. *J. American Statist. Assoc.*, 104(488), 2009.

[DD03] P. Del Moral and A. Doucet. Sequential Monte Carlo samplers. Technical report, Dept. of Engineering, Cambridge Univ., 2003.

[DR10] R. Douc and C.P. Robert. A vanilla variance importance sampling via population Monte Carlo. *Ann. Statist.*, 2010. To appear.

[GBT95] W.R. Gilks, N.G. Best, and K.K.C. Tan. Adaptive rejection Metropolis sampling within Gibbs sampling. *Applied Statist. (Series C)*, 44:455–472, 1995.

[GC10] M. Girolami and B. Calderhead. An object-oriented random-number package with many long streams and substreams. *J. Royal Statist. Society Series B*, 73(2):1–37, 2010.

[GS90] A.E. Gelfand and A.F.M. Smith. Sampling based approaches to calculating marginal densities. *J. American Statist. Assoc.*, 85:398–409, 1990.

[HDL+11] C.C. Holmes, A. Doucet, A. Lee, M. Giles, and C. Yau. Bayesian computation on graphics cards. In J.M. Bernardo, M.J. Bayarri, J.O. Degroot, A.P. Dawid, D. Heckerman, A.M. Smith, and M. West, editors, *Bayesian Statistics 9: Proceedings of the Ninth Valencia International Meeting, June 3-8, 2010*. Oxford University Press, 2011.

[HJPR02] J.P. Hobert, G.L. Jones, B. Presnel, and J.S. Rosenthal. On the applicability of regenerative simulation in Markov chain Monte Carlo. *Biometrika*, 89(4):731–743, 2002.

[LR10] K. Latuszynski and J. S. Rosenthal. Adaptive Gibbs samplers. *ArXiv e-prints*, January 2010.

[LSCK01] Pierre L'Ecuyer, Richard Simard, E. Jack Chen, and W. David Kelton. An object-oriented random-number package with many long streams and substreams. *Operations Research*, 50:1073–1075, 2001.

[LYG+09] A. Lee, C. Yau, M.B. Giles, A. Doucet, and C.C. Holmes. On the utility of graphics cards to perform massively parallel simulation with advanced Monte Carlo methods. *Arxiv preprint arXiv:0905.2441*, 2009.

[MR07] J.-M. Marin and C.P. Robert. *Bayesian Core*. Springer-Verlag, New York, 2007.

[MR10] J.-M. Marin and C.P. Robert. Importance sampling methods for Bayesian discrimination between embedded models. In M.-H. Chen, D.K. Dey, P. Müller, D. Sun, and K. Ye, editors, *Frontiers of Statistical Decision Making and Bayesian Analysis*, pages 513–527. Springer-Verlag, New York, 2010.

[MTY95] P. Mykland, L. Tierney, and B. Yu. Regeneration in Markov chain samplers. *J. American Statist. Assoc.*, 90:233–241, 1995.

[Per99] F. Perron. Beyond accept–reject sampling. *Biometrika*, 86(4):803–813, 1999.

[R D06] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006.

[RC04] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, second edition, 2004.

[Rob95] C.P. Robert. Convergence control techniques for MCMC algorithms. *Statis. Science*, 10(3):231–253, 1995.

[Ros00] J.S. Rosenthal. Parallel computing and Monte Carlo algorithms. *Far East J. Theoretical Statistics*, 4:207–236, 2000.

[SWC+10] M. Suchard, Q. Wang, C. Chan, J. Frelinger, A. Cron, and M. West. Understanding gpu programming for statistical computation: studies in massively parallel massive mixtures. *Journal of Computational and Graphical Statistics*, 19:418–438, 2010.

[WKB+09] D. Wraith, M. Kilbinger, K. Benabed, O. Cappé, J.-F. Cardoso, G. Fort, S. Prunet, and C.P. Robert. Estimation of cosmological parameters using adaptive importance sampling. *Physical Review D*, 80:023507, 2009.

Chapter 3

Free Energy Sequential Monte Carlo

Ce chapitre propose une nouvelle méthode de Monte Carlo séquentiel, appelée Monte Carlo séquentiel à énergie libre. Elle s'inspire d'articles récents parus dans la littérature sur la dynamique des molécules.

La méthode proposée a pour but d'améliorer les performances des méthodes de Monte Carlo lorsque la densité cible est multimodale. Elle nécessite de trouver un axe, appelé "coordonnée de réaction" et noté $\xi(\theta)$ où θ représente la variable aléatoire à simuler. L'axe peut être une fonction de cette variable ou l'une de ses composantes. Il est choisi de manière à ce que, conditionnellement à certaines valeurs de $\xi(\theta)$, la multimodalité de la loi cible soit moins prononcée que celle de la loi initiale, et donc l'exploration de l'espace plus aisée. L'algorithme biaise séquentiellement la loi visée, de manière à ce que les valeurs de $\xi(\theta)$ qui favorisent l'exploration de l'espace soient plus régulièrement visitées que selon la loi originale. À cette fin, l'espace est partitionné selon la coordonnée de réaction, et l'algorithme est défini pour que chaque partie de l'espace soit visitée avec la même fréquence par les particules.

Ainsi, l'échantillon généré est tel que la coordonnée de réaction soit approximativement uniformément distribuée sur un intervalle. Il ne suit donc pas exactement la loi désirée, mais une étape finale d'échantillonnage d'importance permet de s'y ramener. Alternativement, le passage de la loi finale à la loi désirée peut lui-même être effectué par une méthode de Monte Carlo séquentiel.

La méthode est illustrée sur deux exemples de modèles de mélange gaussien: un modèle univarié appliqué à un jeu de données décrivant l'épaisseur de timbres mexicains, classique dans la littérature sur ce genre de modèle, et un modèle bivarié appliqué à des tailles de pétales de fleur, également classique. Les modèles de mélange permettent de représenter graphiquement les performance de la méthode sur un problème fortement multimodal. L'algorithme est comparé à la méthode de Monte Carlo séquentiel classique à coût computationnel égal.

Authors
- Nicolas Chopin (CREST–ENSAE, Paris)
- Pierre E. Jacob (Université Paris-Dauphine, CREST, Paris),

Status Published in *Bayesian Statistics 9*, (proceedings of the ninth Valencia International Meeting on Bayesian Statistics), edited by J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith and M. West, Oxford University Press, 2010.

Note that this chapter is published along with a discussion of the article by Peter Green (University of Bristol) and Benjamin Taylor (University of Lancaster) and with our response to the discussion.

Abstract

We introduce a new class of Sequential Monte Carlo (SMC) methods, which we call free energy SMC. This class is inspired by free energy methods, which originate from Physics, and where one samples from a biased distribution such that a given function $\xi(\theta)$ of the state θ is forced to be uniformly distributed over a given interval. From an initial sequence of distributions (π_t) of interest, and a particular choice of $\xi(\theta)$, a free energy SMC sampler computes sequentially a sequence of biased distributions $(\tilde{\pi}_t)$ with the following properties: (a) the marginal distribution of $\xi(\theta)$ with respect to $\tilde{\pi}_t$ is approximatively uniform over a specified interval, and (b) $\tilde{\pi}_t$ and π_t have the same conditional distribution with respect to ξ. We apply our methodology to mixture posterior distributions, which are highly multimodal. In the mixture context, forcing certain hyper-parameters to higher values greatly facilitates mode swapping, and makes it possible to recover a symmetric output. We illustrate our approach with univariate and bivariate Gaussian mixtures and two real-world datasets.

Keywords: Free energy biasing; Label switching; Mixture; Sequential Monte Carlo; particle filter.

3.1 Introduction

A Sequential Monte Carlo (SMC) algorithm (a.k.a. particle filter) samples iteratively a sequence of probability distributions $(\pi_t)_{t=0,\dots,\mathrm{T}}$, through importance sampling and resampling steps. The initial motivation of SMC was the sequential analysis of dynamic state space models, where π_t stands for the filtering distribution of state (latent variable) x_t, conditional on the data $y_{1:t}$ collected up to time t; see e.g. the book of [Doucet 01]. Recent research however [Neal 01, Chopin 02, Del Moral 06] have extended SMC to "static" problems, which involve a single, but "difficult" (in some sense we detail below) distribution π. Such extensions use an artificial sequence $(\pi_t)_{t=0,\dots,\mathrm{T}}$, starting at some "simple" distribution π_0, and evolving smoothly towards $\pi_T = \pi$. Two instances of such strategies are i) annealing ([Neal 01], see also [Gelman 98]), where $\pi_t(\theta) = \pi_0(\theta)^{1-\gamma_t}\pi(\theta)^{\gamma_t}$, and $\gamma_t = t/T$, or some other increasing sequence that starts at 0 and ends at 1; and ii) IBIS [Chopin 02], where π stands for some Bayesian posterior density $\pi(\theta) = p(\theta|y_{1:T})$, conditional on some complete dataset $y_{1:T}$, and $\pi_t(\theta) = p(\theta|y_{1:t})$. For a general formalism for SMC, see [Del Moral 06].

One typical "difficulty" with distributions of interest π is multimodality. A vanilla sampler typically converges to a single modal region, and fails to detect other modes, which may be of higher density. The two SMC strategies mentioned above alleviate this problem to some extent. In both cases, π_0 is usually unimodal and has a large support, so "particles" (sampled points) explore the sample space freely during the first iterations. However, this initial exploration is not always sufficient to prevent the sample from degenerating to a single modal region. We give an illustration of this point in this paper.

To overcome multimodality, the molecular dynamics community has developed in recent years an interesting class of methods, based on the concept of free energy biasing; see for instance the book of [Lelièvre 10] for a general introduction. Such methods assume the knowledge of a low-dimensional function $\xi(\theta)$, termed as the "reaction coordinate", such that, conditional on $\xi(\theta) = x$, the multimodality (a.k.a. metastability in the physics literature) of π is much less severe, at least for certain values of x. The principle is then to sample from $\tilde{\pi}$, a free energy biased version of π, $\tilde{\pi}(\theta) = \pi(\theta) \exp\{A \circ \xi(\theta)\}$, where A denotes the free energy, that is, minus log the marginal density of the random variable $\xi(\theta)$, with respect to π. This forces an uniform exploration of the random variable $\xi(\theta)$, within given bounds. At a final stage, one may perform importance sampling from $\tilde{\pi}$ to π to recover the true distribution π.

The main difficulty in free energy biasing methods is to estimate the free energy A. A typical approach is to compute sequentially an estimate $A_{(t)}$ of A, using some form of Adaptive MCMC (Markov chain Monte Carlo): at each iteration t, a MCMC step is performed, which leaves invariant $\pi_{(t)}(\theta) = \pi(\theta) \exp\left\{A_{(t)} \circ \xi(\theta)\right\}$, then a new estimate $A_{(t+1)}$ of the free energy is computed from the simulated process up to time t. The simulation is stopped when the estimate $A_{(t)}$ stabilises in some sense. Convergence of Adaptive MCMC samplers is a delicate subject: trying to learn too quickly from the past may prevent convergence for instance. These considerations are outside the scope of this paper, and we refer the interested reader to the review by [Andrieu 08] and references therein.

Instead, our objective is to bring the concept of free energy biasing to the realm of SMC. Specifically, and starting from some pre-specified sequence (π_t), we design a class of SMC samplers, which compute sequentially the free energy A_t associated to each distribution π_t, and track the sequence of biased densities $\tilde{\pi}_t(\theta) = \pi_t(\theta) \exp\{A_t \circ \xi(\theta)\}$. In this way, particles may move freely between the modal regions not only at the early iterations, where π_t remains close to π_0 and therefore is not strongly multimodal, but also at the later stages, thanks to free energy biasing.

We apply free energy SMC sampling to the Bayesian analysis of mixture models. [Chopin 10] show that free energy biasing methods are an interesting approach for dealing with the multimodality of mixture posterior distributions. In particular, they present several efficient reaction coordinates for univariate Gaussian mixtures, such as the hyper-parameter that determines the prior expectation of the component variances. In this paper, we investigate how free energy SMC compares with this initial approach based on Adaptive MCMC, and to which extent such ideas may be extended to other mixture models, such as a bivariate Gaussian mixture model.

The paper is organised as follows. Section 3.2 describes the SMC methodology. Section 3.3 presents the concept of free energy biased sampling. Section 3.4 presents a new class of SMC methods, termed as free energy SMC. Section 3.5 discusses the application to Bayesian inference for mixtures, and presents numerical results, for two types of mixtures (univariate Gaussian, bivariate Gaussian), and two datasets. Section 3.6 concludes.

3.2 SMC algorithms

3.2.1 Basic structure

In this section, we describe briefly the structure of SMC algorithms. For the sake of exposition, we consider a sequence of probability densities π_t, $t = 0, \ldots, T$ defined on a common space $\Theta \subset \mathbb{R}^d$. At each iteration t, a SMC algorithm produces a weighted sample $(w_{t,n}, \theta_{t,n})$, $n = 1, \dots, N$, which targets π_t in the following sense:

$$\frac{\sum_{n=1}^{N} w_{t,n}\varphi(\theta_{t,n})}{\sum_{n=1}^{N} w_{t,n}} \rightarrow_{N\rightarrow+\infty} \mathbb{E}^{\pi_t}\left\{\varphi(\theta)\right\},$$

almost surely, for a certain class of test functions φ. At iteration 0, one typically samples $\theta_{0,n} \sim \pi_0$, and set $w_{0,n} = 1$. To progress from iteration $t-1$ to iteration t, it is sufficient to perform a basic importance sampling step from π_{t-1} to π_t:

$$\theta_{t,n} = \theta_{t-1,n}, \quad w_{t,n} = w_{t-1,n} \times u_t(\theta_{t,n})$$

where u_t denotes the incremental weight function

$$u_t(\theta) = \frac{\pi_t(\theta)}{\pi_{t-1}(\theta)}.$$

However, if only importance sampling steps are performed, the algorithm is equivalent to a single importance sampling step, from π_0 to π_T. This is likely to be very inefficient. Instead, one should regularly perform resample-move steps [Gilks 01], that, is, a succession of i) a resampling step, where current points $\theta_{t,n}$ are resampled according to their weights, so that points with a small (resp. big) weight are likely to die (resp. generate many offsprings); and ii) a mutation step, where each resampled point is "mutated" according to some probability kernel $K_t(\theta, d\hat{\theta})$, typically a MCMC kernel with invariant distribution π_t. In the more general formalism of [Del Moral 06], this is equivalent to performing an importance sampling step in the space $\Theta \times \Theta$, with forward kernel K_t, associated to some probability density $K_t(\theta, \hat{\theta})$, and backward kernel L_t associated to the probability density $L_t(\hat{\theta}, \theta) = \pi_t(\theta)K_t(\theta, \hat{\theta})/\pi_t(\hat{\theta})$.

Resample-move steps should be performed whenever the weight degeneracy is too high. A popular criterion is $\mathrm{EF}(t) < \tau$, where $\tau \in (0, 1)$, and EF is the efficency factor, that is the effective sample size of [Kong 94] divided by N,

$$\mathrm{EF}(t) = \frac{\left(\sum_{n=1}^{N} w_{t,n}\right)^2}{N\sum_{n=1}^{N} w_{t,n}^2}.$$

We summarise in Algorithm 1 the general structure of SMC algorithms. There are several methods for resampling the particles, e.g. the multinomial scheme [Gordon 93], the residual scheme [Liu 98], the systematic scheme [Whitley 94, Carpenter 99]. We shall use the systematic scheme in our simulations.

3.2.2 Adaptiveness of SMC

In contrast to MCMC, where designing adaptive algorithms require a careful convergence study, designing adaptive SMC samplers is straightforward. We consider first the design of the MCMC kernels K_t. For instance, [Chopin 02] uses independent Hastings-Metropolis kernels, with a Gaussian proposal fitted to the current particle sample. This is a reasonable strategy if π_t is close to Gaussianity. In this paper, we consider instead the following strategy, which seems more generally applicable: take K_t as a succession of k Gaussian random walk Hastings-Metropolis steps $K_{t,i}(\theta, d\theta')$, i.e. simulating from $K_{t,i}(\theta, d\theta')$ consists

Algorithm 3 A generic SMC algorithm

```
0. Sample $\theta_{0,n} \sim \pi_0$, set $w_{0,n} = 1$, for $n = 1, \ldots, N$. Set $t = 1$.
1. Compute new weights as
       $w_{t,n} = w_{t-1,n} \times u_t(\theta_{t-1,n})$.
2. If $\mathrm{EF}(t) < \tau$, then
   (a) resample the particles, i.e. construct a sample $(\hat{\theta}_{t,n})_{1 \leq n \leq N}$ made of $R_{t,n}$ replicates of
       particle $\theta_{t,n}$, $1 \leq n \leq N$, where $R_{t,n}$ is a nonnegative integer-valued random variable such
       that
           $\mathbb{E}\left[R_{t,n}\right] = \frac{N w_{t,n}}{\sum_{n'=1}^{N} w_{t,n'}}$,
       and set $w_{t,n} = 1$.
   (b) move the particles with respect to Markov kernel $K_t$,
           $\theta_{t,n} \sim K_t(\hat{\theta}_{t,n}, d\theta)$
   otherwise
           $\theta_{t,n} = \theta_{t-1,n}$.
3. $t \leftarrow t + 1$, if $t < T$ go to Step 1.
```

of proposing a value $\theta' \sim N_d(\theta, \Sigma_{t,i})$, accepting this value with probability $1 \wedge \{\pi(\theta')/\pi(\theta)\}$, otherwise keep the current value θ. Then take $\Sigma_{t,i} = c_{t,i} S_t$, $c_{t,i} > 0$, and S_t is the empirical covariance matrix of the resampled particles at iteration t (that is, the particles obtained immediately before the MCMC step with kernel K_t is performed). The constant $c_{t,i}$ may be tuned automatically as well. For instance, one may start with $c_0 = 0.3$, then, each time the acceptance rate of the MCMC step is below (resp. above) a given threshold, the constant c_t is divided (resp. multiplied) by two.

As in MCMC, it is common to focus on the adaptiveness of the transition kernels, but one may use the particle sample (or the history of the process in the MCMC context) to adapt the target distributions as well. This is precisely what we do in this paper, since the target at time t on our free energy SMC sampler shall depend on a bias function which is estimated from the current particle sample, see Section 3.4.

3.2.3 IBIS versus annealing, choice of π_0

When the distribution of interest π is some Bayesian posterior density

$$\pi(\theta) = p(\theta | y_{1:D}) = \frac{1}{Z} p(\theta) p(y_{1:D} | \theta),$$

where $y_{1:D}$ is a vector of D observations, $p(\theta)$ is the prior density, and $p(y_{1:D}|\theta)$ is the likelihood, it is of interest to compare the two aforementioned SMC strategy, namely,

1. IBIS, where $T = D$, and $\pi_t(\theta) = p(\theta|y_{1:t})$, in particular, $\pi_0(\theta) = p(\theta)$ is the prior; and

2. Annealing, where $\pi_t(\theta) = \pi_0(\theta)^{1-\gamma_t} \pi(\theta)^{\gamma_t}$, γ_t is an increasing sequence such that $\gamma_0 = 0$, and $\gamma_T = 1$, π_0 is typically the prior density, but could be something else, and T and D do not need to be related.

Clearly, for the same number of particles, and assuming that the same number of resample-move steps is performed, IBIS is less time-consuming, because calculations at iteration t involve only the t first observations. On the other hand, annealing may produce a smoother sequence of distributions, so it may require less resample-move steps. [Jasra 07] provide numerical examples where the IBIS strategy leads to unstable estimates. In the context discussed in the paper, see Section 3.5, and elsewhere, we did not run into cases where IBIS is particularly unstable. Perhaps it is fair to say that a general comparison is not meaningful, as the performance of both strategies seems quite dependent on the applications, and also various tuning parameters such as the sequence γ_t for instance.

We take this opportunity however to propose a simple method to improve the regularity of the IBIS sequence, in the specific case where the observations are exchangeable and real-valued. We remark first that this regularity depends strongly on the order of incorporation of the y_t's. For instance, sorting the observations in ascending order would certainly lead to very poor performance. On the other hand, a random order would be more suitable, and was recommended by [Chopin 02]. Pushing this idea further, we propose the following strategy: First, we re-define the median of a sample as either the usual median, when D is an odd number, or the smallest of the two middle values in the ordered sample, when D is an even number. Then, we take y_1 as the median observation, y_2 (resp. y_3) to be the median of the observations that are smaller (resp. larger) than y_1, then we split again the four corresponding sub-samples by selecting some values y_4 to y_7, and so on, until all values are selected. We term this strategy as “Van der Corput ordering”, as a Van der Corput binary sequence is precisely defined as $1/2$, $1/4$, $3/4$, $1/8, \ldots$

A point which is often overlooked in the literature, and which affects both strategies, is the choice of π_0. Clearly, if $\pi_0(\theta) = p(\theta)$, one may take the prior so uninformative that the algorithm degenerates in one step. Fortunately, in the application we discuss in this paper, namely Bayesian analysis of mixture models, priors are often informative; see Section 3.5 for a discussion of this point. In other contexts, it may be helpful to perform a preliminary exploration of π in order design some π_0, quite possibly different from the prior, so that (i) for the annealing strategy, moving from π_0 to $\pi_T = \pi$ does not take too much time; and (ii) for the IBIS strategy, one can use π_0 as an artificial prior, and recover the prior of interest at the final stage of the algorithm, by multiplying all the particle weights by $p(\theta)/\pi_0(\theta)$.

3.3 Free energy-biased sampling

3.3.1 Definition of free energy and free-energy biased densities

In this section we explain in more detail the concept of free energy biasing. We consider a single distribution of interest, defined by a probability density π with respect to the Lebesgue measure associated to $\Theta \subset \mathbb{R}^d$. As explained in the introduction, the first step in implementing a free energy biasing method is to choose a reaction coordinate, that is, some measurable function $\xi : \theta \to \mathbb{R}^{d'}$, where d' is small. In this paper, we take $d' = 1$. One assumes that the multimodality of π is strongly determined, in some sense, by the direction $\xi(\theta)$. For instance, the distribution of θ, conditional on $\xi(\theta) = x$, may be much less multimodal than the complete distribution π, for either all or certain values of x.

In words, the free energy is, up to an arbitrary constant, minus the logarithm of the marginal density of $\xi(\theta)$. The free energy may be written down informally as

$$\exp\left\{-A(x)\right\} \propto \int_\Theta \pi(\theta) \mathrm{I}_{[x,x+dx]}\left\{\xi(\theta)\right\} d\theta$$

and more rigorously, as

$$\exp\left\{-A(x)\right\} \propto \int_{\Omega_x} \pi(\theta)\, d\left\{\theta|\xi(\theta)=x\right\},$$

where $\Omega_x = \{\theta \in \Theta : \xi(\theta) = x\}$, and $d\{\theta|\xi(\theta) = x\}$ denotes the conditional measure on the set Ω_x which is "compatible" with Lebesgue measure on the embedding space Θ, i.e. volumes are preserved and so on. In both formulations, the proportionality relation indicates that the density π may be known only up to a multiplicative constant, and therefore that the free energy is defined only up to an arbitrary additive constant.

The free energy biased density $\tilde{\pi}$ is usually defined as

$$\tilde{\pi}(\theta) \propto \pi(\theta) \exp\left\{A \circ \xi(\theta)\right\} \mathrm{I}_{[x_{\min}, x_{\max}]}\left\{\xi(\theta)\right\}$$

where $[x_{\min}, x_{\max}]$ is some pre-defined range and $A \circ \xi(\theta) = A(\xi(\theta))$. It is clear that, with respect to $\tilde{\pi}$, the marginal distribution of the random variable $\xi(\theta)$ is uniform over $[x_{\min}, x_{\max}]$, and the conditional distributions of θ, given $\xi(\theta) = x$ matches the same conditional distribution corresponding to π. Figure 3.1–3.3 illustrate free energy biasing on a toy example. Figure 3.1 shows the original bimodal target distribution and the choice of the reaction coordinate. Figure 3.2 represents the corresponding free energy. Figure 3.3 finally represents the free energy biased density. The objective is to sample from $\tilde{\pi}$, which requires to estimate the free energy A.

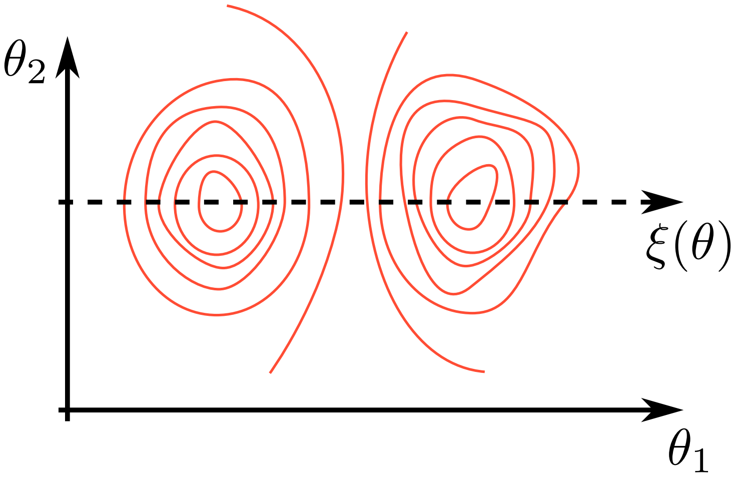


Figure 3.1: Contour plot of some bimodal distribution defined on $\mathbb{R}^2$. The modes are horizontally aligned, hence we choose $\xi(\theta) = \theta_1$.

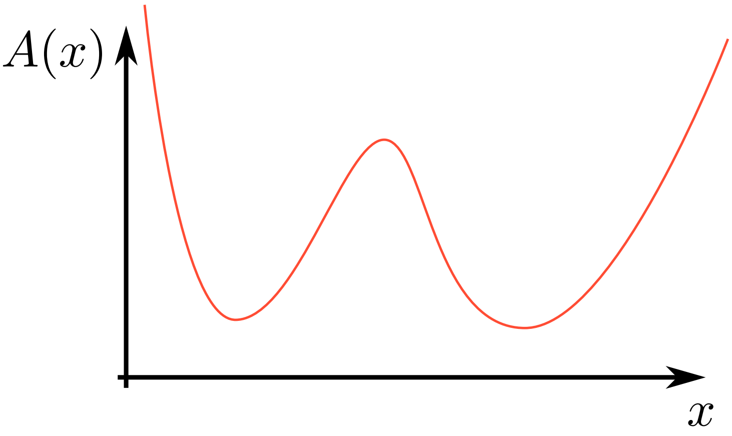


Figure 3.2: Shape of the free energy A for the target distribution plotted on Figure 3.1. Here x corresponds to θ_1, and $A \circ \xi(\theta)$ could be written $A(\theta_1)$.

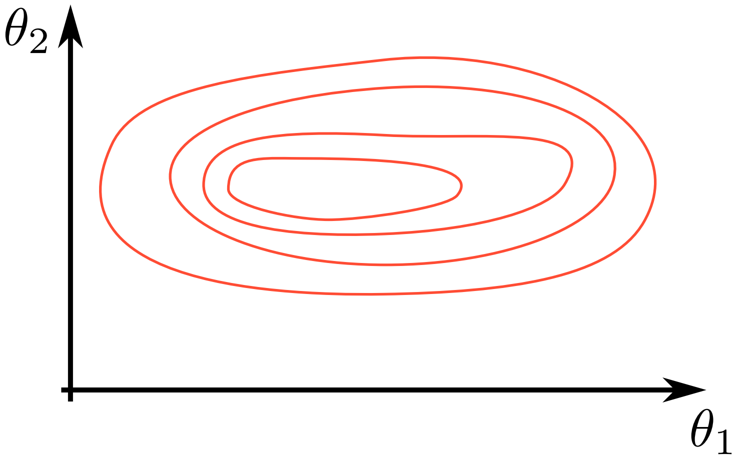


Figure 3.3: Contour plot of the free energy biased density corresponding to the original target represented on Figure 3.1 and the free energy represented on Figure 3.2.

To avoid the truncation incurred by the restriction to interval $[x_{\min}, x_{\max}]$, we shall consider instead the following version of the free-energy biased density $\tilde{\pi}_t$:

$$\tilde{\pi}(\theta) \propto \pi(\theta) \exp \{A \circ \xi(\theta)\}$$

where the definition of A is extended as follows: $A(x) = A(x_{\min})$ for $x \leq x_{\min}$, $A(x) = A(x_{\max})$ for $x \geq x_{\max}$.

3.3.2 Estimation of the free energy

As explained in the introduction, one usually resorts to some form of Adaptive MCMC to estimate the free energy A. Specifically, one performs successive MCMC steps (typically Hastings-Metropolis), such that the Markov kernel $K_{(t)}$ used at iteration t depends on the trajectory of the simulated process up to time $t-1$. (The simulated process is therefore non-Markovian.) The invariant distribution of kernel $K_{(t)}$ is $\pi_{(t)}(\theta) \propto \pi(\theta) \exp \left\{A_{(t)} \circ \xi(\theta)\right\}$, where $A_{(t)}$ is an estimate of the free energy A that has been computed at iteration t, from the simulated trajectory up to time $t-1$. Note that the brackets in the notations $K_{(t)}$, $\pi_{(t)}$, $A_{(t)}$ indicate that all these quantities are specific to this section and to the Adaptive MCMC context, and must not mistaken for the similar quantities found elsewhere in the paper, such as, e.g. the density π_t targeted at iteration t by a SMC sampler. The difficulty is then to come up with an efficient estimator (or rather a sequence of estimators, $A_{(t)}$), of the free energy.

Since this paper is not concerned with adaptive MCMC, we consider instead the much simpler problem of estimating the free energy A from a weighted sample $(\theta_n, w_n)_{n=1,\dots,N}$ targeting π; for instance, the θ_n's could be i.i.d. with probability density g, and $w_n = \pi(\theta)/g(\theta)$. Of course, this discussion is simplistic from an Adaptive MCMC perspective, but it will be sufficient in our SMC context. We refer the reader to e.g. [Chopin 10] for the missing details.

First, it is necessary to discretise the problem, and consider some partition:

$$[x_{\min}, x_{\max}] = \cup_{i=0}^{n_x}[x_i, x_{i+1}], \quad x_i = x_{\min} + (x_{\max} - x_{\min})\frac{i}{n_x}. \tag{3.1}$$

Then, they are basically two ways to estimate A. The first method is to estimate directly a discretised version of A, by simply computing an estimate the proportion of points that fall in each bin:

$$\exp\left\{-\hat{A}_1(x)\right\} = \frac{\sum_{n=1}^{N} w_n \mathrm{I}\left\{\xi(\theta_n) \in [x_i, x_{i+1}]\right\}}{\sum_{n=1}^{N} w_n}, \quad \text{for } x \in [x_i, x_{i+1}].$$

The second method is indirect, and based on the following property: the derivative of the free energy is such that

$$A'(x) = \mathbb{E}^{\pi}\left[f(\theta)|\xi(\theta) = x\right]$$

where the force f is defined as:

$$f = -\frac{(\nabla \log \pi)\cdot(\nabla \xi)}{|\nabla \xi|^2} - \operatorname{div}\left(\frac{\nabla \xi}{|\nabla \xi|^2}\right),$$

and ∇ (resp. div) is the gradient (resp. divergence) operator. Often, $\xi(\theta)$ is simply a coordinate of the vector θ, $\theta = (\xi, \dots)$, in which case the expression above simplifies to $f = -\partial \log \pi / \partial \xi$. This leads to the following estimator of the derivative of A:

$$\hat{A}_2'(x) = \frac{\sum_{n=1}^{N} w_n \mathrm{I}\left\{\xi(\theta_n) \in [x_i, x_{i+1}]\right\} f(\theta_n)}{\sum_{n=1}^{N} w_n \mathrm{I}\left\{\xi(\theta_n) \in [x_i, x_{i+1}]\right\}}, \quad \text{for } x \in [x_i, x_{i+1}].$$

Then an estimate of A may be deduced by simply computing cumulative sums for instance:

$$\hat{A}_2(x) = \sum_{j: x_j \le x} \hat{A'}_2(x_j)(x_{j+1} - x_j), \quad \text{for } x \in [x_i, x_{i+1}].$$

Methods based on the first type of estimates are usually called ABP (Adaptive Biasing Potential) methods, while methods of the second type are called ABF (Adaptive Biasing Force). Empirical evidence suggests that ABF leads to slightly smoother estimates, presumably because it is based on a derivative.

3.4 Free energy SMC

We now return to the SMC context, and consider a pre-specified sequence (π_t). Our objective is to derive a SMC algorithm which sequentially computes the free energy A_t

associated to each density π_t,

$$\exp\left\{-A_t(x)\right\} \propto \int \pi_t(\theta) d\left\{\theta | \xi(\theta) = x\right\}$$

and sample $\tilde{\pi}_t$, the free energy biased version of π_t,

$$\tilde{\pi}_t(\theta) \propto \pi_t(\theta) \exp\left\{A_t \circ \xi(\theta)\right\}.$$

Again, to avoid truncating to interval $[x_{\min}, x_{\max}]$, one extends the definition of A_t outside $[x_{\min}, x_{\max}]$ by taking $A_t(x) = A_t(x_{\min})$ for $x < x_{\min}$, $A_t(x) = A_t(x_{\max})$ for $x > x_{\max}$.

As explained in Section 3.3.2, one actually estimates a discretised version of the free energy, i.e., the algorithm shall provide estimates $\hat{A}_t(x_i)$, $i = 0, \ldots, n_x$ of the free energy evaluated at grid points over an interval $[x_{\min}, x_{\max}]$, as defined in (3.1). Note that this grid is the same for all iterations t.

Assume that we are at the end of iteration $t-1$, that estimates $\hat{A}_{t-1}(x_i)$ of A_{t-1} have been obtained, and that the particle system $(\theta_{t-1,n}, w_{t-1,n})_{n=1,\ldots,N}$ targets $\tilde{\pi}_{t-1}$. If the particles are re-weighted according to the incremental weight function $u_t(\theta) = \pi_t(\theta)/\pi_{t-1}(\theta)$, i.e.

$$\bar{w}_{t,n} = w_{t-1,n} \times u_t(\theta_{t-1,n})$$

then the new target distribution of the particle system $(\theta_{t-1,n}, \bar{w}_{t,n})_{n=1,\ldots,N}$ is

$$\bar{\pi}_t(\theta) \propto \tilde{\pi}_{t-1}(\theta) u_t(\theta).$$

The objective is then to recover $\tilde{\pi}_t$, which depends on the currently unknown free energy A_t. To that effect, we first state the following result.

Theorem 1. *The free energy D_t associated to $\bar{\pi}_t$ is*

$$D_t = A_t - A_{t-1}$$

that is, the difference between the free energies of π_t and π_{t-1}.

Proof. One has, for $\theta \in \Theta$,

$$\bar{\pi}_t(\theta) \propto \pi_t(\theta) \exp\left\{A_{t-1} \circ \xi(\theta)\right\}$$

hence, for $x \in \xi(\Theta)$,

$$\int_{\Omega_x} \bar{\pi}_t(\theta) d\left\{\theta | \xi(\theta) = x\right\} = \exp\left\{\left(A_{t-1} - A_t\right)(x)\right\}.$$

and one concludes. □

This result provides the justification for the following strategy. First, particles are reweighted from π_{t-1} to $\bar{\pi}_t$, as explained above. Second, the free energy D_t of $\bar{\pi}_t$ is estimated, using either the ABP or the ABF strategy, see Section 3.3.2; this leads to some estimate $\hat{D}_t$ of D_t, ot more precisely estimates $\hat{D}_t(x_i)$ over the grid $x_0, \ldots, x_{n_x}$. From this, one readily obtains estimates of the current free energy, using the proposition above:

$$\hat{A}_t(x_i) = \hat{A}_{t-1}(x_i) + \hat{D}_t(x_i), \quad i = 0, \ldots, n_x. \tag{3.2}$$

Third, one recovers $\tilde{\pi}_t$ by performing an importance sampling step from $\bar{\pi}_t$ to $\tilde{\pi}_t$; this is equivalent to updating the weights as follows:

$$w_{t,n} = \bar{w}_{t,n} \exp\left\{\hat{D}_t \circ \xi(\theta_{t,n})\right\}.$$

An outline of this free energy SMC algorithm is given in Algorithm 2.

```
Algorithm 4 Free energy SMC
0. Sample $\theta_{0,n} \sim \pi_0$, set $w_{0,n} = 1$, for $n = 1, \dots, N$. Compute $A_0$ and set $t = 1$.
1. Compute new weights as
       $\bar{w}_{t,n} = w_{t-1,n} \times u_t(\theta_{t-1,n})$.
2. Compute an estimator $\hat{D}_t$ of free energy $D_t$, compute weights
       $w_{t,n} = \bar{w}_{t,n} \exp\left[\hat{D}_t \circ \xi(\theta_{t-1,n})\right]$
   and update the estimate $\hat{A}_t$ of the free energy $A_t$, using (3.2).
3. If $\mathrm{EF}(t) < \tau$, then
   (a) resample the particles, i.e. draw randomly $\hat{\theta}_{t,n}$ in such a way that
       $\mathbb{E}\left[\sum_{n'=1}^{N} \mathrm{I}(\hat{\theta}_{t,n'} = \theta_{t-1,n}) \,\middle|\, (\theta_{t-1,n}, w_{t,n})\right] = \frac{N w_{t,n}}{\sum_{n'=1}^{N} w_{t,n'}}$
       and set $w_{t,n} = 1$.
   (b) move the particles with respect to Markov kernel $K_t$,
       $\theta_{t,n} \sim K_t(\hat{\theta}_{t,n}, d\theta)$
   otherwise
       $\theta_{t,n} = \theta_{t-1,n}$.
4. $t \leftarrow t + 1$, if $t < T$ go to Step 1.
```

At the final stage of the algorithm (iteration T), one recovers the unbiased target $\pi_T = \pi$ by a direct importance sampling step, from $\tilde{\pi}_T$ to π_T:

$$\frac{\pi_T(\theta)}{\tilde{\pi}_T(\theta)} \propto \exp\left\{\hat{A}_T \circ \xi(\theta)\right\}.$$

This is because of this ultimate debiasing step, which relies on $\hat{A}_T$, that one must store in memory and compute iteratively the "complete" free energy A_T (as opposed to the successive D_t, which may be termed as "incremental" free energies). If this unbiasing step is too "brutal", meaning that too many particles get a low weight in the final sample, then one may apply instead a progressive unbiasing strategy, by extending the sequence of distributions $\tilde{\pi}_T$ as follows:

$$\tilde{\pi}_{T+l}(\theta) \propto \tilde{\pi}_T(\theta) \exp\left\{\left(\frac{l}{L}\right) \hat{A}_T \circ \xi(\theta)\right\}, \quad l = 0, \dots, L$$

and performing additional SMC steps, that is, successive importance sampling steps from $\tilde{\pi}_{T+l}$ to $\tilde{\pi}_{T+l+1}$, and, when necessary, resample-move steps in order to avoid degeneracy. In our simulations, we found that progressive unbiasing did lead to some improvement, but that often direct unbiasing was sufficient. Hence, we report only results from direct unbiasing in the next Section.

3.5 Application to mixtures

3.5.1 General formulation, multimodality

A $K-$component Bayesian mixture model consists of D independent and identically distributed observations y_i, with parametric density

$$p(y_i|\theta) = \frac{1}{\sum_{k=1}^K \omega_k} \sum_{k=1}^K \omega_k \psi(y_i; \xi_k), \quad \omega_k \geq 0,$$

where $\{\psi(\cdot;\xi),\ \xi \in \Xi\}$ is some parametric family, e.g. $\psi(y,\xi) = N(y;\mu,1/\lambda)$, $\xi = (\mu,\lambda^{-1})$. The parameter vector contains

$$\theta = (\omega_1, \ldots, \omega_k, \xi_1, \ldots, \xi_k, \eta),$$

where η is the set of hyper-parameters that are shared by the K components. The prior distribution $p(\theta)$ is typically symmetric with respect to component permutation. In particular, one may assume that, a priori and independently $\omega_k \sim \text{Gamma}(\delta, 1)$. This leads to a $\text{Dirichlet}_K(\delta,\ldots,\delta)$ prior for the component probabilities

$$q_k = \frac{\omega_k}{\sum_{l=1}^K \omega_l}, \quad k = 1, \ldots K.$$

We note in passing that, while the formulation of a mixture model in terms of the $q_k's$ is more common, we find that the formulation in terms of the unnormalised weights ω_k is both more tractable (because it imposes symmetry in the notations) and more convenient in terms of implementation (e.g. designing Hastings-Metropolis steps).

An important feature of the corresponding posterior density

$$\pi(\theta) = p(\theta|y_{1:D}) \propto p(\theta) \prod_{i=1}^D p(y_i|\theta),$$

assuming D observations are available, is its invariance with respect to "label permutation". This feature and its bearings to Monte Carlo inference have received a lot of attention, see e.g. [Celeux 00], [Jasra 05], [Chopin 10] among others. In short, a standard MCMC sampler, such as the Gibbs sampler of [Diebolt 94], see also the book of [Frühwirth-Schnatter 06], typically visits a single modal region. But, since the posterior is symmetric, any mode admits $K! - 1$ replicates in Θ. Therefore, one can assert that the sampler has not converged. [Frühwirth-Schnatter 01] proposes to permute randomly the components at each iteration. However, [Jasra 05] mentions the risk of "genuine multimodality", that is, the $K!$ symetric modal regions visited by the permutation sampler may still represent a small part of the posterior mass, because other sets of equivalent modes have not been visited. [Marin 07, Chap. 6] and [Chopin 10] provide practical examples of this phenomenon.

One could say that random permutations merely "cure the most obvious symptom" of failed convergence. We follow [Celeux 00], [Jasra 05] and [Chopin 10], and take the opposite perspective that one should aim at designing samplers that produce a nearly symmetric output (with respect to label switching), *without resorting to random permutations.*

3.5.2 Univariate Gaussian mixtures

Prior, reaction coordinates

We first consider a univariate Gaussian mixture model, i.e.

$$\psi(y, \xi) = N(y; \mu, \lambda^{-1})$$

and $\xi = (\mu, \lambda^{-1})$, and we use the same prior as in [Richardson 97], that is, for $k = 1, \dots, K$, independently,

$$\mu_k \sim N(M, \kappa^{-1}), \quad \lambda_k \sim \text{Gamma}(\alpha, \beta),$$

where α, M and κ are fixed, and β is a hyper-parameter:

$$\beta \sim \text{Gamma}(g, h).$$

Specifically, we take $\delta = 1$, $\alpha = 2$ (see Chap. 6 of Frühwirth-Schnatter 06 for a justification), $g = 0.2$, $h = 100g/\alpha R^2$, $M = \bar{y}$, and $\kappa = 4/R^2$, where $\bar{y}$ and R are, respectively, the empirical mean and the range of the observed sample.

Regarding the application of free energy methods to univariate Gaussian mixture posterior distributions, [Chopin 10] find that the two following functions of θ are efficient reaction coordinates: $\xi(\theta) = \beta$, and the potential function $V(\theta) = -\log\{p(\theta)p(y_{1:D}|\theta)\}$, that is, up to a constant, minus log the posterior density. However, the latter reaction coordinate is less convenient, because it is difficult to determine in advance the range $[x_{\min}, x_{\max}]$ of exploration. This is even more problematic in our sequential context. Using the IBIS strategy for instance, one would define $V_t(\theta) = -\log\{p(\theta)p(y_{1:t}|\theta)\}$, but the range of likely values for V_t would typically be very different between small and large values of t. Thus we discard this reaction coordinate.

In constrast, as discussed already in [Chopin 10], it is reasonably easy to determine a range of likely values for the reaction coordinate $\xi(\theta) = \beta$. In our simulations, we take $[x_{\min}, x_{\max}] = [R^2/2000, R^2/20]$, where, again, R is the range of the data. [Chopin 10] explains the good performance of this particular reaction coordinate as follows. Large values of β penalise small component variances, thus forcing β to large values leads to a conditional posterior distribution which favours overlapping components, which may switch more easily.

Numerical example

We consider the most challenging example discussed in [Chopin 10], namely the Hidalgo stamps dataset, see e.g. [Izenman 88] for details, and $K = 3$. In particular, [Chopin 10] needed about 10^9 iterations of an Adaptive MCMC sampler (namely, an ABF sampler) to obtain a stable estimate of the free energy.

We run SMC samplers with the following settings: the number of particles is $N = 2 \times 10^4$, the criterion for triggering resample-move steps is ESS$< 0.8N$, and a move step consists of 10 successive Gaussian random walk steps, using the automatic calibration strategy described in Section 3.2.2.

We first run a SMC sampler, without free energy biasing, and using the IBIS strategy. Results are reported in Figures 3.4 and 3.5: the output is not symmetric with respect to label permutation, and only one modal region of the posterior distribution is visited.

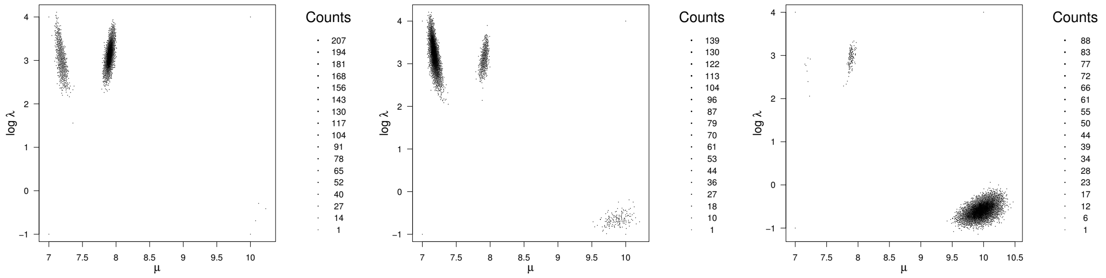


Figure 3.4: Hexagon binning for $(\mu_k, \log\lambda_k)$, $k = 1, 2, 3$, for the standard SMC sampler, no free energy biasing, IBIS strategy.

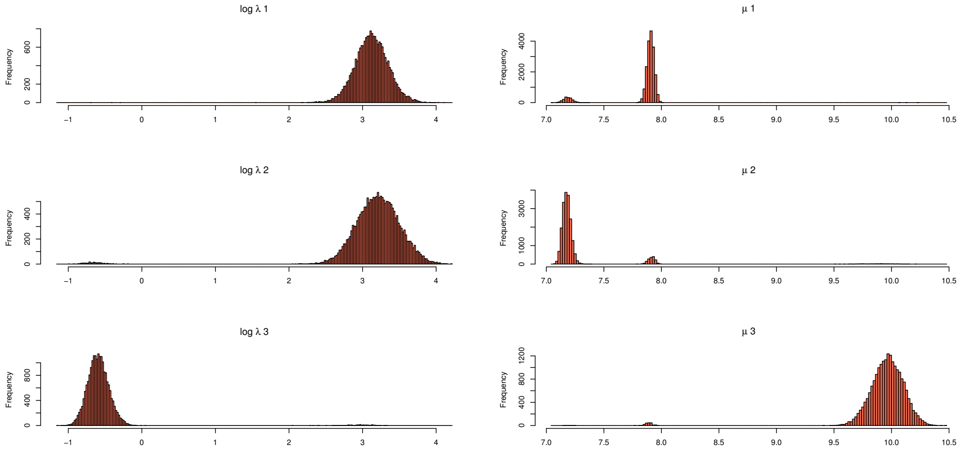


Figure 3.5: Weighted 1D histograms for the standard SMC sampler, no free energy biasing, IBIS strategy.

We then run a free energy SMC sampler, using the reaction coordinate ξ, 50 bins, and the ABP strategy for estimating the free energies. Figures 3.6 and 3.7 represent the cloud of particles before the final unbiasing step, when the particles target the free energy biased density $\tilde{\pi}_T$. Figures 3.8 and 3.9 represent the cloud of particles at the final step, when the target is the true posterior distribution. One sees that the output is not perfectly symmetric (at least after the final debiasing step), but at least the three equivalent modes have been recovered, and one can force equal proportions for the particles in each modal region, by simply randomly permuting the labels of each particles, if need be.

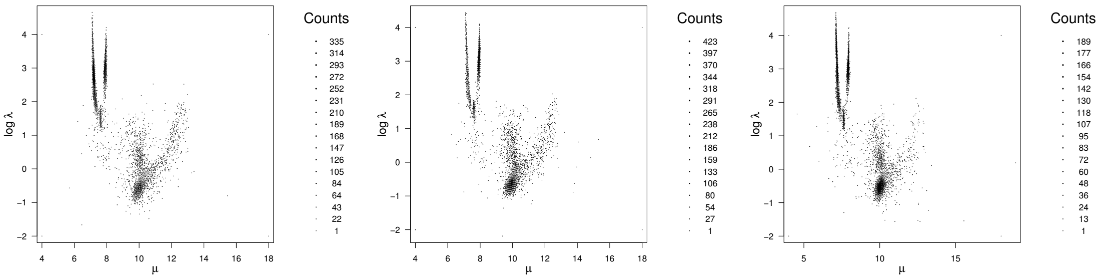


Figure 3.6: Hexagon binning for $(\mu_k, \log\lambda_k)$, $k = 1, 2, 3$, for the free energy SMC sampler, before the final debiasing step, IBIS strategy.

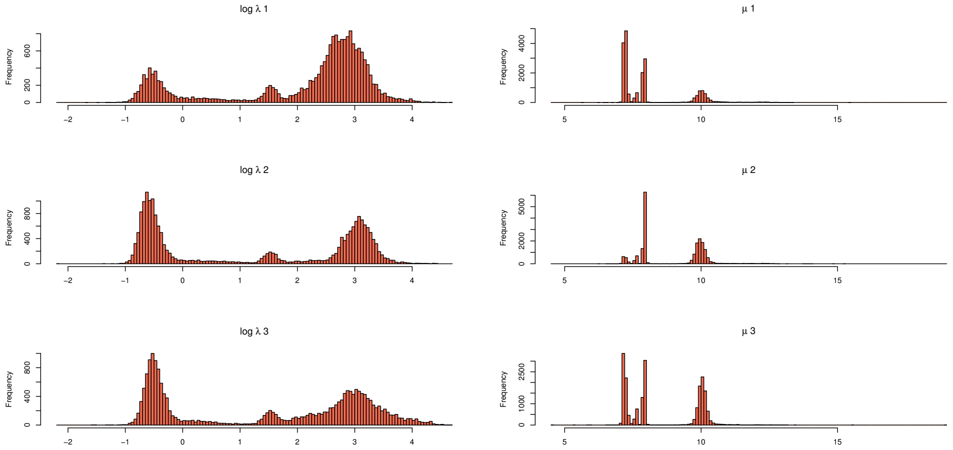


Figure 3.7: Histograms of the components of the simulated particles obtained by free energy SMC sampler, before the final debiasing step, IBIS strategy.

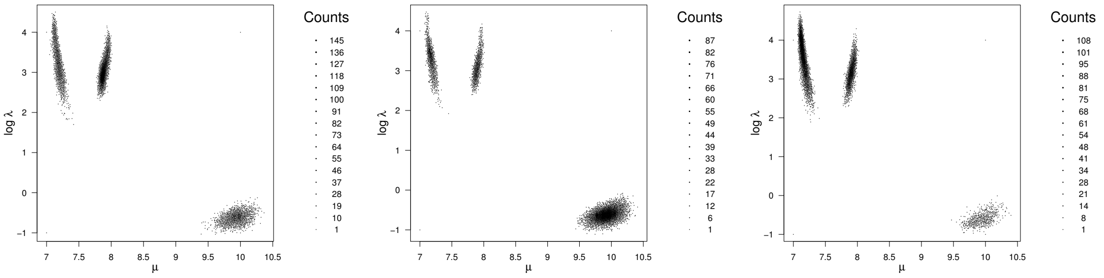


Figure 3.8: Hexagon binning for $(\mu_k, \log\lambda_k)$, $k = 1, 2, 3$, for the free energy SMC sampler, after the final debiasing step, IBIS strategy.

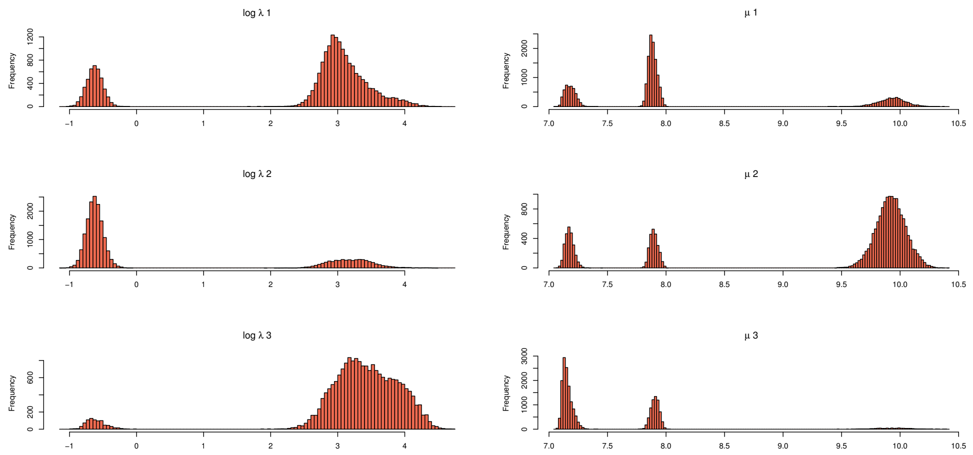


Figure 3.9: Histograms of the components of the simulated particles obtained by free energy SMC sampler, after the final debiasing step, IBIS strategy.

To assess the stability of our results, we run the same sampler ten times, and plot the ten so-obtained estimates of the overall free energy A_T, which is used in the last debiasing step; see Figure 3.10. Since a free energy function is defined only up to an additive function, we arbitrarily force the plotted functions to have the same minimum.

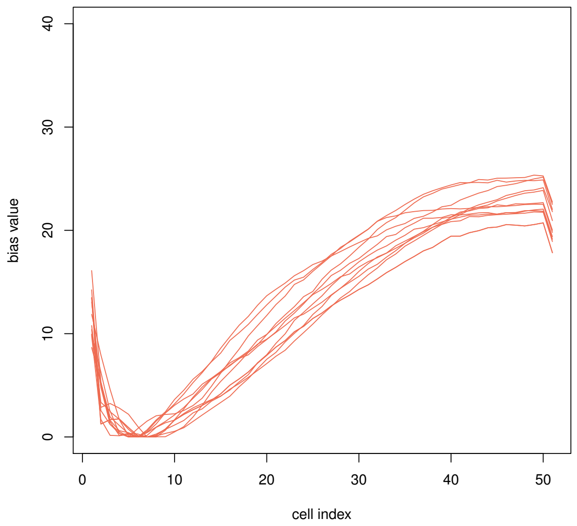


Figure 3.10: Estimates of the final free energy A_T obtained from 10 runs of a free energy SMC sampler, versus cell indices

In short, one sees in this challenging example that (a) a nearly symmetric output is obtained only if free energy biasing is implemented; and (b) using free energy SMC, satisfactory results are obtained at a smaller cost than the adaptive MCMC sampler used in [Chopin 10].

3.5.3 Bivariate Gaussian mixtures

Prior, reaction coordinates

We now consider a bivariate Gaussian mixture, $\psi(y;\xi) = N_2\left(\mu, Q^{-1}\right)$, with the following parameters:

$$\xi_k = \left(\mu_{1,k}, \mu_{2,k}, d_{1,k}, d_{2,k}, e_k\right), \quad C_k = \begin{pmatrix} d_{1,k}^{1/2} & 0 \\ e_k & d_{2,k}^{1/2} \end{pmatrix}, \quad Q_k = C_k C_k^T.$$

This parametrisation is based on Bartlett decomposition: taking

$$\begin{aligned} d_{1,k} &\sim \text{Gamma}(\alpha/2, \beta) \\ d_{2,k} &\sim \text{Gamma}((\alpha-1)/2, \beta) \\ e_k|\beta &\sim N(0, 1/\beta) \end{aligned}$$

leads to a Wishart prior for Q_k, $Q_k \sim \text{Wishart}_2(\alpha, \beta I_2)$. This parametrisation is also convenient in terms of implementing the automatically tuned random walk Hastings-Metropolis strategy discussed in Section 3.2.2.

To complete the specification of the prior, we assume that

$$\mu_k = (\mu_{1,k}, \mu_{2,k})' \sim N_2\left(M, S^{-1}\right),$$

that $\alpha = 2$, and that $\beta \sim \text{Gamma}(g, h)$. Of course, this prior is meant to generalise the prior used in the previous section in a simple way. In particular, the hyper-parameter β should play the same role as in the univariate Gaussian case, and we use it as our reaction coordinate.

Numerical results

We consider two out of the four measurements recorded in Fisher's Iris dataset, petal length and petal width, see e.g. [Frühwirth-Schnatter 06, Chap. 6], and Figure 3.11 for a scatter-plot. We take $K = 2$. As in the previous example, we run a standard SMC sampler (with the same number of particles, and so on), and observes that only one mode is recovered. We then run a free energy SMC sampler. For the sake of space, we report only the debiased output at the very final stage of the free energy SMC sampler, that is the cloud of particles targeting the true posterior distribution. Figure 3.12 represents the bivariate vectors μ_k, and Figure 3.13 represent the component probabilities $q_k = \omega_k/(\omega_1 + \omega_2)$ for $k = 1$, 2. Clearly, the output is nearly symmetric.

One sees in this example that free energy SMC still works well for bivariate Gaussian mixture model, despite the larger dimension of the parameter space. In particular, the choice of the reaction coordinate seems to work along the same lines, i.e. choosing an hyper-parameter that determines the spread of the components.

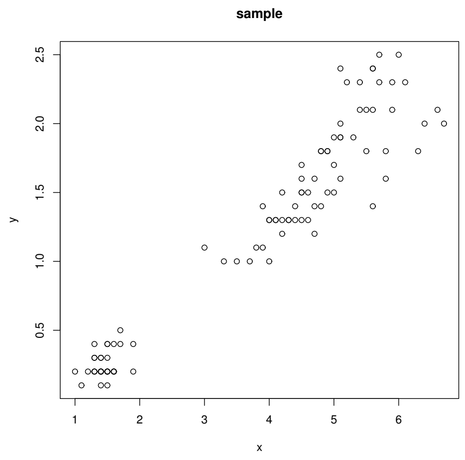


Figure 3.11: Iris sample

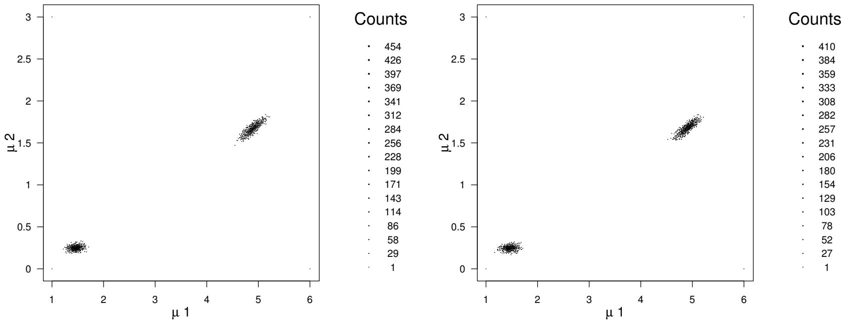


Figure 3.12: Hexagon binning for $\mu_k = (\mu_{k,1}, \mu_{k,2})$, $k = 1$, bivariate Gaussian example

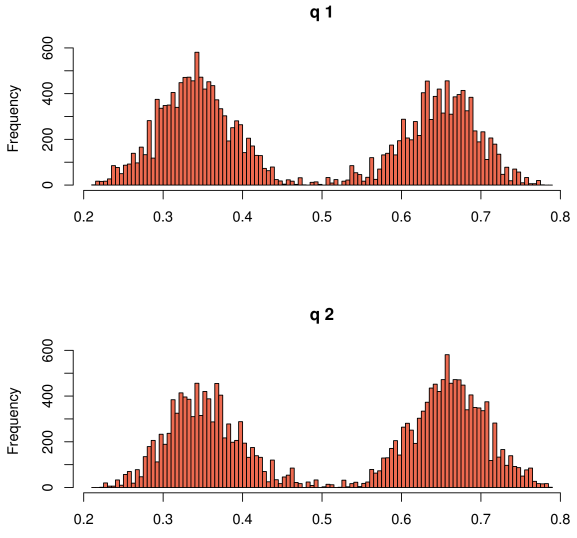


Figure 3.13: Weighted histograms of $q_k = \omega_k/(\omega_1 + \omega_2)$, for $k = 1$, 2, bivariate Gaussian example

3.6 Conclusion

In this paper, we introduced free energy SMC sampling, and observed in one mixture example that it may be faster than free energy methods based on adaptive MCMC, such as those considered in [Chopin 10]. It would be far-fetched to reach general conclusions from this preliminary study regarding the respective merits of free energy SMC versus free energy MCMC, or, worse, SMC versus Adaptive MCMC. If anything, the good results obtained in our examples validates, in the mixture context, the idea of combining two recipes to overcome multimodality, namely (a) free energy biasing, and (b) tracking through SMC some sequence (π_t) of increasing difficulty, which terminates at $\pi_T = \pi$. Whether such combination should work or would be meaningful in other contexts is left for further research.

Acknowledgements

N. Chopin is supported by the 2007–2010 grant ANR-07-BLAN “SP Bayes”. P. Jacob is supported by a PhD Fellowship from the AXA Research Fund. The authors thank Peter Green for insightful comments.

Bibliography

[Andrieu 08] C. Andrieu & J. Thoms. *A tutorial on adaptive MCMC.* Statist. Comput., vol. 18, no. 4, pages 343–373, 2008.

[Carpenter 99] J. Carpenter, P. Clifford & P. Fearnhead. *Improved Particle Filter for nonlinear problems.* IEE Proc. Radar, Sonar Navigation, vol. 146, no. 1, pages 2–7, 1999.

[Celeux 00] G. Celeux, M. Hurn & C. P. Robert. *Computational and inferential difficulties with mixture posterior distributions.* J. Am. Statist. Assoc., vol. 95, pages 957–970, 2000.

[Chopin 02] N. Chopin. *A sequential particle filter for static models.* Biometrika, vol. 89, pages 539–552, 2002.

[Chopin 10] N. Chopin, T. Lelievre & G. Stoltz. *Free energy methods for efficient exploration of mixture posterior densities.* Arxiv preprint arXiv:1003.0428, 2010.

[Del Moral 06] P. Del Moral, A. Doucet & A. Jasra. *Sequential Monte Carlo samplers.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 68, no. 3, pages 411–436, 2006.

[Diebolt 94] J. Diebolt & C.P. Robert. *Estimation of finite mixture distributions through Bayesian sampling.* J. R. Statist. Soc. B, pages 363–375, 1994.

[Doucet 01] A. Doucet, N. de Freitas & N. J. Gordon. Sequential Monte Carlo methods in practice. Springer-Verlag, New York, 2001.

[Frühwirth-Schnatter 01] S. Frühwirth-Schnatter. *Markov chain Monte Carlo estimation of classical and dynamic switching and mixture models.* J. Am. Statist. Assoc., vol. 96, no. 453, pages 194–209, 2001.

[Frühwirth-Schnatter 06] S. Frühwirth-Schnatter. Finite mixture and markov switching models. Springer, 2006.

[Gelman 98] A. Gelman & X.L. Meng. *Simulating normalizing constants: From importance sampling to bridge sampling to path sampling.* Statistical Science, vol. 13, no. 2, pages 163–185, 1998.

[Gilks 01] W. R. Gilks & C. Berzuini. *Following a moving target - Monte Carlo inference for dynamic Bayesian models.* J. R. Statist. Soc. B, vol. 63, pages 127–146, 2001.

[Gordon 93] N. J. Gordon, D. J. Salmond & A. F. M. Smith. *Novel approach to nonlinear/non-Gaussian Bayesian state estimation.* IEE Proc. F, Comm., Radar, Signal Proc., vol. 140, no. 2, pages 107–113, 1993.

[Izenman 88] A. J. Izenman & C. J. Sommer. *Philatelic mixtures and multimodal densities.* J. Am. Statist. Assoc., no. 83, pages 941–953, 1988.

[Jasra 05] A. Jasra, CC Holmes & DA Stephens. *Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling.* Statist. Science, pages 50–67, 2005.

[Jasra 07] A. Jasra, D.A. Stephens & C.C. Holmes. *On population-based simulation for static inference.* Statistics and Computing, vol. 17, no. 3, pages 263–279, 2007.

[Kong 94] A. Kong, J. S. Liu & W. H. Wong. *Sequential imputation and Bayesian missing data problems.* J. Am. Statist. Assoc., vol. 89, pages 278–288, 1994.

[Lelièvre 10] T. Lelièvre, M. Rousset & G. Stoltz. Free-energy computations: a mathematical perspective. Imperial College Press, 2010.

[Liu 98] J. Liu & R. Chen. *Sequential Monte Carlo methods for dynamic systems.* J. Am. Statist. Assoc., vol. 93, pages 1032–1044, 1998.

[Marin 07] J.M. Marin & C.P. Robert. Bayesian core: a practical approach to computational Bayesian statistics. Springer Verlag, 2007.

[Neal 01] R. M. Neal. *Annealed importance sampling.* Statist. Comput., vol. 11, pages 125–139, 2001.

[Richardson 97] S. Richardson & P. J. Green. *On Bayesian analysis of mixtures with an unknown number of components.* J. R. Statist. Soc. B, vol. 59, no. 4, pages 731–792, 1997.

[Whitley 94] Whitley. *A genetic algorithm tutorial.* Statist. Comput., vol. 4, pages 65–85, 1994.

Chapter 4

Wang–Landau and the Flat Histogram criterion

L'algorithme de Wang–Landau fait partie de la classe des algorithmes de Monte Carlo à chaîne de Markov adaptatifs. Il est conçu pour être robuste face à l'éventuelle multimodalité de la loi cible, de manière similaire à la méthode présentée dans le chapitre 3. Ce chapitre étudie une variante de l'algorithme et établit sa validité théorique.

Plus précisément, l'algorithme génère un échantillon $(X_t)_{t\geq 0}$ selon une succession d'étapes de Metropolis–Hastings, telle qu'à l'étape t la loi visée, notée π_t, soit une modification de la loi cible originale π. Cette modification favorise les régions de l'espace qui n'ont pas été encore assez visitées par la chaîne $(X_t)_{t\geq 0}$. Plus formellement, une partition de l'espace en $d \in \mathbb{N}$ régions est introduite, notée $(\mathcal{X}_i)_{i=1}^d$, ainsi qu'un vecteur $(\phi_i)_{i=1}^d$ représentant les fréquences de visite désirées pour chaque région. L'algorithme est défini de sorte que les fréquences de visite de la chaîne dans chaque région convergent vers les fréquences désirées.

Ce chapitre établit cette convergence, au sens suivant. On dit que le critère "Flat Histogram" est atteint lorsque la différence maximale entre les fréquences observées et les fréquences désirées est inférieure à un seuil fixé au préalable. Ce chapitre propose une preuve que, quelque soit le seuil fixé, le critère "Flat Histogram" est vérifié en un temps dont l'espérance est finie. Cette preuve établit la validité d'une variante de l'algorithme de Wang–Landau qui utilise le critère "Flat Histogram" pour calibrer la stabilisation du caractère adaptatif de l'algorithme. Ce faisant, elle établit aussi l'invalidité d'une autre variante de l'algorithme. Les hypothèses fortes sous lesquelles la preuve est établie sont discutées, puis les conséquences du résultat sont illustrées sur un exemple jouet.

Ce chapitre présente donc une étude sur une méthode de Monte Carlo à chaîne de Markov adaptative, qui ne vérifie pas la condition usuelle d'adaptation appelée *diminishing adaptation*, selon laquelle les paramètres de l'algorithme convergent vers des valeurs fixées. Par ailleurs il s'agit d'un algorithme où la loi de proposition est fixée, mais la loi visée change à chaque étape. Ces deux éléments contribuent à l'originalité de l'algorithme et de l'étude proposée dans ce chapitre.

Authors
- Pierre E. Jacob (Université Paris-Dauphine, CREST, Paris),
- Robin J. Ryder (Université Paris-Dauphine, CREST, Paris),

Status Technical report, arXiv number 1110.4025.

Abstract

The Wang-Landau algorithm aims at sampling from a probability distribution, while penalizing some regions of the state space and favouring others. It is widely used, but its convergence properties are still unknown. We show that for some variations of the algorithm, the Wang-Landau algorithm reaches the so-called Flat Histogram criterion in finite time, and that this criterion can be never reached for other variations. The arguments are shown on an simple context – compact spaces, density functions bounded from both sides– for the sake of clarity, and could be extended to more general contexts.

Keywords: Markov Chain Monte Carlo, Wang-Landau algorithm, Flat Histogram.

4.1 Introduction and notations

Consider the problem of sampling from a probability distribution π defined on a measure space $(\mathcal{X}, \Sigma, \mu)$. We suppose that we can compute the probability density function of π at any point $x \in \mathcal{X}$, up to a multiplicative constant. Given a proposal kernel $Q(\cdot, \cdot)$ we define a Metropolis–Hastings (MH) [Hastings 70, Tierney 98] transition kernel targeting π, denoted by $K(\cdot, \cdot)$, as follows:

$$\forall x, y \in \mathcal{X} \quad K(x, y) = Q(x, y)\rho(x, y) + \delta_x(y)(1 - r(x))$$

with $\rho(x, y)$ defined by $\rho(x, y) := 1 \wedge \frac{\pi(y)}{\pi(x)} \frac{Q(y,x)}{Q(x,y)}$ and $r(x)$ defined by:

$$r(x) := \int_{\mathcal{X}} \rho(x, y)Q(x, y)dy$$

Here the delta function $\delta_a(b)$ takes value 1 when $a = b$ and 0 otherwise. Under some conditions on the proposal Q and the target π, the MH kernel defines an algorithm to generate a Markov chain with stationary distribution π [Robert 04].

Let us consider a partition of the state space $\mathcal{X}$ into d disjoint sets $\mathcal{X}_1, \ldots, \mathcal{X}_d$:

$$\mathcal{X} = \bigcup_{i=1}^{d} \mathcal{X}_i$$

If we have a sample $X_1, \ldots, X_t$ independent and identically distributed from π, then for any $i \in [1, d]$:

$$\frac{1}{t} \sum_{n=1}^{t} \mathbb{I}_{\mathcal{X}_i}(X_n) \xrightarrow[t \to \infty]{\mathbb{P}} \int_{\mathcal{X}_i} \pi(x)dx =: \psi_i$$

where we denote by $\mathbb{I}_{\mathcal{X}_i}(x)$ the indicator function that is equal to 1 when $x \in \mathcal{X}_i$ and 0 otherwise. Similar convergence is obtained when $X_1, \ldots, X_t$ is an ergodic chain such as the one generated by the MH algorithm. The purpose of the Wang-Landau algorithm [Wang 01a, Wang 01b, Liang 05, Atchadé 10] is to obtain a sample

- such that for any $i \in [1, d]$ the subsample
$$\{X_n \text{ for } n \in [1, t] \text{ s.t. } X_n \in \mathcal{X}_i\}$$
is distributed according to the restriction of π to $\mathcal{X}_i$, and

- such that for any $i \in [1, d]$
$$\frac{1}{t} \sum_{n=1}^{t} \mathbb{I}_{\mathcal{X}_i}(X_n) \xrightarrow[t\to\infty]{\mathbb{P}} \phi_i$$
where $\phi = (\phi_1, \dots, \phi_d)$ is chosen by the user, and could be any vector in $]0, 1[^d$ such that $\sum_{i=1}^{d} \phi_i = 1$.

A typical use of this algorithm is to sample from multimodal distributions, by penalizing already-visited regions and favouring the exploration of regions between modes, in an attempt to recover all the modes.

This algorithm, in the class of Markov Chain Monte Carlo (MCMC) algorithms [Robert 04], therefore allows to learn about π while “forcing” the proportions of visits ϕ_i of the generated chain to any of the sets $\mathcal{X}_i$, which are typically also chosen by the user. The vector $\phi_1, \dots, \phi_d$ might be referred to as the “desired frequencies”, and the sets $\mathcal{X}_i$ are called the “bins”. In a typical situation, the mass of π over bin $\mathcal{X}_i$, which we denote by ψ_i, is unknown, and hence one cannot easily guess how much to “penalize” or to “favour” a bin $\mathcal{X}_i$ in order to obtain the desired frequency ϕ_i. The Wang-Landau algorithm introduces a vector $\theta_t = (\theta_t(1), \dots, \theta_t(d))$, referred to as “penalties” at time t, which is updated at every iteration t, and which acts like an approximation of the ratios $\psi_1/\phi_1, \dots, \psi_d/\phi_d$, up to a multiplicative constant.

For a distribution π and a vector of penalties $\theta = (\theta(1), \dots, \theta(d))$, we define the penalized distribution π_θ:
$$\pi_\theta(x) \propto \pi(x) \times \sum_{i=1}^{d} \frac{\mathbb{I}_{\mathcal{X}_i}(x)}{\theta(i)}$$
To be more concise we define a function $J : \mathcal{X} \mapsto \{1, \dots, d\}$ that takes a state $x \in \mathcal{X}$ and returns the index i of the bin $\mathcal{X}_i$ such that $x \in \mathcal{X}_i$. We can now write: $\pi_\theta(x) \propto \pi(x)/\theta(J(x))$. We will denote by K_θ the MH kernel targeting π_θ.

The Wang-Landau algorithm, described in the next section, alternates between generating a sample by targeting π_θ using K_θ, and updating θ using the generated sample. In this sense it is an adaptive MCMC algorithm (past samples are used to update the kernel at a given iteration), using an auxiliary chain (θ_t), and therefore the behaviour of the sample is not obvious.

The Wang-Landau algorithm is widely used in the Physics community, see for example [Silva 06, Malakis 06, Cunha Netto 06]. In particular, many practitioners use flavours of the algorithm with a “Flat-Histogram” criterion. However, its convergence properties are still partially unknown. We show that this criterion is reached in finite time for some variations of the algorithm. This result is all that was missing to apply results on adaptive algorithms with diminishing adaptation [Fort 11].

In Section 4.2, we define variations of the Wang-Landau algorithm. We then introduce ratios of penalties and argue for their convenience in studying the properties of the algorithm. We prove in Sections 4.3 and 4.4 that under certain conditions, the Flat Histogram criterion is met in finite time, for the cases $d = 2$ and $d > 2$ respectively. The result is illustrated in Section 4.5, and in Section 4.6, we hint at how our assumptions might be relaxed.

4.2 Wang-Landau algorithms: different flavours

There are several versions of the Wang-Landau algorithm. We describe the general version introduced by [Atchadé 10], both in its deterministic form and with a stochastic schedule.

4.2.1 A first version with deterministic schedule

Let $(\gamma_t)_{t\in\mathbb{N}}$ (referred to as a schedule or a temperature) be a sequence of positive real numbers such that:

$$\begin{cases} \sum_{t\geq 0}\gamma_t & = \infty \\ \sum_{t\geq 0}\gamma_t^2 & < \infty \end{cases}$$

A typical choice is $\gamma_t := t^{-\alpha}$ with $\alpha \in]0.5, 1[$. The Wang-Landau algorithm is described in pseudo-code in Algorithm 5. In this form, the schedule γ_t decreases at each iteration, and is therefore called "deterministic".

Algorithm 5 Wang-Landau with deterministic schedule

```
1: Init $\forall i \in \{1,\dots,d\}$ set $\theta_0(i) \leftarrow 1/d$.
2: Init $X_0 \in \mathcal{X}$.
3: for $t = 1$ to $T$ do
4:     Sample $X_t$ from $K_{\theta_{t-1}}(X_{t-1},\cdot)$, MH kernel targeting $\pi_{\theta_{t-1}}$.
5:     Update the penalties: $\log\theta_t(i) \leftarrow \log\theta_{t-1}(i) + f(\mathbb{I}_{\mathcal{X}_i}(X_t), \phi_i, \gamma_t)$.
6: end for
```

Step 5 of Algorithm 5 updates the penalties from θ_{t-1} to θ_t, by increasing it if the corresponding bin has been visited by the chain at the current iteration, and by decreasing it otherwise. This rationale seems natural, however we did not find any article arguing for a particular choice of update, among the infinite number of updates that would also follow the same rationale. In other words, it is not obvious how to choose the function f, except that it should be such that it is positive when $X_t \in \mathcal{X}_i$ and such that it is closer to 0 when γ_t decreases, to ensure that the penalties converge. Some practitioners use the following update:

$$\log\theta_t(i) \leftarrow \log\theta_{t-1}(i) + \gamma_t\left(\mathbb{I}_{\mathcal{X}_i}(X_t) - \phi_i\right) \tag{4.1}$$

while others use:

$$\log\theta_t(i) \leftarrow \log\theta_{t-1}(i) + \log\left[1 + \gamma_t\left(\mathbb{I}_{\mathcal{X}_i}(X_t) - \phi_i\right)\right] \tag{4.2}$$

Since γ_t converges to 0 when t increases, and since update (4.1) is the first-order Taylor expansion of update (4.2), one legitimately expects both updates to result in similar performance in practice. We shall see in Section 4.3 that this is not necessarily the case.

Some convergence results have been proven about Algorithm 5: the deterministic schedule ensures that θ_t changes less and less along the iterations of the algorithm, and consequently the kernels K_{θ_t} change less and less as well. The study of the algorithm hence falls into the realm of adaptive MCMC where the *diminishing adaptation* condition holds [Andrieu 08, Atchadé 09, Fort 11], although it is original in the sense that the target distribution (π_{θ_t}) is adaptive but not necessarily the proposal distribution Q. See also the literature on stochastic approximation [Andrieu 06].

In this article we are especially interested in a more sophisticated version of the Wang-Landau algorithm that uses a stochastic schedule, and for which, as we shall see in the following, the two updates result in different performance.

4.2.2 A sophisticated version with stochastic schedule

A remarkable improvement has been made over Algorithm 5: the use of a “Flat Histogram” (FH) criterion to decrease the schedule only at certain random times. Let us introduce $\nu_t(i)$, the number of generated points at iteration t that are in $\mathcal{X}_i$:

$$\nu_t(i) := \sum_{n=1}^{t} \mathbb{1}_{\mathcal{X}_i}(X_n)$$

For some predefined precision threshold c, we say that (FH) is met at iteration t if:

$$\max_{i\in\{1,\ldots,d\}} \left| \frac{\nu_t(i)}{t} - \phi_i \right| < c$$

Intuitively, this criterion is met if the observed proportion of visits to each bin is not far from ϕ, the desired proportion. The name “Flat Histogram” comes from the observation that if the desired proportions are all equal to $1/d$, this criterion is verified when the histogram of visits is approximately flat. The threshold c could possibly decrease along the iterations, to get an always finer precision.

The Wang-Landau with Flat Histogram (Algorithm 6) is similar to the previous algorithm, with a single difference: the schedule γ does not decrease at each step anymore, but only when (FH) is met. To know whether it is met or not, a counter ν_t of visits to each bin is updated at each iteration, and when (FH) is met, the schedule decreases and the counter is reset to 0.

Algorithm 6 Wang-Landau with Flat Histogram

```
 1: Init $\forall i \in \{1,\ldots,d\}$ set $\theta_0(i) \leftarrow 1/d$.
 2: Init $X_0 \in \mathcal{X}$.
 3: Init $\kappa = 0$, the number of (FH) criteria already reached.
 4: Init the counter $\forall i \in \{1,\ldots,d\}$   $\nu_1(i) \leftarrow 0$
 5: for $t = 1$ to $T$ do
 6:     Sample $X_t$ from $K_{\theta_{t-1}}(X_{t-1},\cdot)$ targeting $\pi_{\theta_{t-1}}$.
 7:     Update $\nu_t$: $\forall i \in \{1,\ldots,d\}$   $\nu_t(i) \leftarrow \nu_{t-1}(i) + \mathbb{1}_{\mathcal{X}_i}(X_t)$
 8:     Check whether (FH) is met.
 9:     if (FH) is met then
10:         $\kappa \leftarrow \kappa + 1$
11:         $\forall i \in \{1,\ldots,d\}$   $\nu_t(i) \leftarrow 0$
12:     end if
13:     Update the bias: $\log\theta_t(i) \leftarrow \log\theta_{t-1}(i) + f(\mathbb{1}_{\mathcal{X}_i}(X_t), \phi_i, \gamma_\kappa)$.
14: end for
```

Note the difference between Algorithms 5 and 6: γ is indexed by κ instead of t, and κ is a random variable. As with Algorithm 5, the update of penalties (step 13 of Algorithm 6) can be either update (4.1) or update (4.2), or possibly something else. Interestingly in this case, it is not obvious anymore that both updates will give similar results. Indeed, for γ_κ to go to 0, we need (FH) to be reached in finite time, so that κ regularly increases.

This flavour of the Wang-Landau algorithm is widely used in the Physics literature [Cunha Netto 06, Silva 06, Malakis 06, Ngo 08].

Our contribution is to show in a simple context that update (4.1) is such that (FH) is met in finite time, while (4.2) is not so. Hence only using update (4.1) can one expect the convergence properties of Algorithm 5 to still hold for Algorithm 6, since if (FH) is met in finite time a sort of *diminishing adaptation* condition would still hold.

To underline the difficulty of knowing whether (FH) is met in finite time or not, let us recall that between two (FH) occurrences, the schedule is constant (equal to some $\gamma_\kappa > 0$), hence the penalties (θ_t) change at a constant scale and *diminishing adaptation* does not directly hold. Other adaptive algorithms share this lack of *diminishing adaptation*, as e.g. the Accelerated Stochastic Approximation algorithm [Kesten 58], in which the adaptation of some process (X_t) diminishes only if its increments change sign. In our case, (FH) will be reached if the chain (X_t) lands with frequency ϕ_i in each bin $\mathcal{X}_i$ (see Corollary 2).

Note that in the implementation of the algorithm, the penalties θ_t need only be defined up to a normalizing constant, since they only appear in ratios of the form $\theta_t(i)/\theta_t(j)$. We therefore introduce the following notation:

$$\forall i,j \in \{1,\ldots,d\} \text{ such that } i \neq j \quad Z_t^{(i,j)} = \log\frac{\theta_t(i)}{\theta_t(j)}$$

and we note Z_t the collection of all the $Z_t^{(i,j)}$. Some intuition behind the study of such ratios comes from considering update (4.1). With this update, assume that for each i, $\mathbb{E}[\mathbb{1}_{\mathcal{X}_i}(X_t)] = \phi_i$. Then we could easily check that for each pair (i,j), $\mathbb{E}[Z_t^{(i,j)}|Z_{t-1}^{(i,j)}] = Z_{t-1}^{(i,j)}$, so this process would be constant on average. The remainder of this paper hinges on two facts: that we can control (Z_t), in the sense that $Z_t^{(i,j)}/t \to 0$; and that if we control (Z_t), then we control the frequencies of visits (ν_t/t).

More generally, notice that with fixed γ, the pair (X_t, Z_t) forms a homogeneous Markov chain. If we could prove that this chain is irreducible, then it would imply that its proportion of visits to the set $\mathcal{X}_i \times \mathbb{R}^{d(d-1)}$ converges to some value in $[0,1]$. We would then need to check that the limit is indeed the desired frequency ϕ_i for all i. Unfortunately, properties of the joint chain (X_t, Z_t) are difficult to establish due to the complexity of its transition kernel. Finding a so-called drift function for the joint Markov chain is also typically difficult. In general, we are not able to show that the chain is irreducible. In section 4.3, we prove directly that $Z_t^{1,2}/t \to 0$ in the special case $d = 2$, under some assumptions. In section 4.4, we make more restrictive assumptions which imply irreducibility. In both cases, we show the implication of this convergence on the frequencies of visits.

4.3 Proof when $d = 2$

In the following we consider a simple context with only two bins: $d = 2$ and X_t can therefore only be either in $\mathcal{X}_1$ or in $\mathcal{X}_2$. Suppose the current schedule is at $\gamma > 0$, and we want to know whether (FH) is going to be met in finite time (hence γ is fixed here). To simplify notation, in this section we note

$$Z_t = Z_t^{(1,2)} = \log\theta_t(1) - \log\theta_t(2)$$

Using the definition of the penalties (θ_t) and of the counts (ν_t), we obtain

$$\begin{aligned}
Z_t &= Z_0 + [\nu_t(1)f(1,\phi_1,\gamma) + (t-\nu_t(1))f(0,\phi_1,\gamma)] \\
&\quad - [\nu_t(2)f(1,\phi_2,\gamma) + (t-\nu_t(2))f(0,\phi_2,\gamma)] \\
&= Z_0 + \nu_t(1)\,[f(1,\phi_1,\gamma) - f(0,\phi_1,\gamma)] + tf(0,\phi_1,\gamma) \\
&\quad - [(t-\nu_t(1))f(1,\phi_2,\gamma) + \nu_t(1)f(0,\phi_2,\gamma)] \\
&= Z_0 + \nu_t(1)\,[f(1,\phi_1,\gamma) - f(0,\phi_1,\gamma) + f(1,\phi_2,\gamma) - f(0,\phi_2,\gamma))] \\
&\quad + t(f(0,\phi_1,\gamma) - f(1,\phi_2,\gamma))
\end{aligned}$$

If we prove that Z_t/t goes to 0 (for instance in mean), this will imply the following convergence of the proportion of visits:

$$\frac{\nu_t(1)}{t} \xrightarrow[t\to\infty]{} \frac{f(1,\phi_2,\gamma) - f(0,\phi_1,\gamma)}{f(1,\phi_1,\gamma) - f(0,\phi_1,\gamma) + f(1,\phi_2,\gamma) - f(0,\phi_2,\gamma)}$$

(also in mean). Since we want (FH) to be reached in finite time for any precision threshold $c > 0$, we need the proportions of visits to $\mathcal{X}_i$ to converge to ϕ_i. Hence we want:

$$\frac{f(1,\phi_2,\gamma) - f(0,\phi_1,\gamma)}{f(1,\phi_1,\gamma) - f(0,\phi_1,\gamma) + f(1,\phi_2,\gamma) - f(0,\phi_2,\gamma)} = \phi_1 \tag{4.3}$$

Using the specific forms of $f(\mathbb{1}_{\mathcal{X}_i}(X_t),\phi_i,\gamma)$ for both updates, we can easily see that

- update (4.1) satisfies equation (4.3) for any ϕ and γ;
- in general, update (4.2) does not satisfy equation (4.3), except in the special case where $\phi_1 = \phi_2 = 1/2$.

The rest of the paper is devoted to the proof that Z_t/t goes to 0 under some assumptions. More formally, we state in Theorem 1 what we shall prove in the remainder of this section. This theorem holds for both updates.

Theorem 1. *Consider the sequence of penalties (θ_t) introduced in Algorithm 6. We define:*

$$Z_t = \log\theta_t(1) - \log\theta_t(2)$$

Then:

$$\frac{Z_t}{t} \xrightarrow[t\to\infty]{L_1} 0$$

As a consequence, the long run proportion of visits to each bin converges to the desired frequency ϕ for update (4.1), and not necessarily for update (4.2). Corollary 2 clarifies the consequence of Theorem 1 on the validity of Algorithm 6.

Corollary 2. *When the proportions of visits converge in mean to the desired proportions, the Flat Histogram criterion is reached in finite time for any precision threshold c.*

We already made the simplification of considering the simple case $d = 2$. We make the following assumptions:

Assumption 1. *The bins are not empty with respect to μ and π:*

$$\forall i \in \{1,2\} \quad \mu(\mathcal{X}_i) > 0 \text{ and } \pi(\mathcal{X}_i) > 0$$

Assumption 2. *The state space $\mathcal{X}$ is compact.*

Assumption 3. *The proposition distribution $Q(x, y)$ is such that:*

$$\exists q_{min} > 0 \quad \forall x \in \mathcal{X} \quad \forall y \in \mathcal{X} \quad Q(x, y) > q_{min}$$

Assumption 4. *The MH acceptance ratio is bounded from both sides:*

$$\exists m > 0 \quad \exists M > 0 \quad \forall x \in \mathcal{X} \quad \forall y \in \mathcal{X} \quad m < \frac{\pi(y)}{\pi(x)} \frac{Q(y, x)}{Q(x, y)} < M$$

Assumption 1 guarantees that the bins are well designed, and if it was not verified, the algorithm would never reach (FH), regardless of the other assumptions. Assumptions 2-4 are for example verified by a gaussian random walk proposal over a compact space, where there is a lower bound on π. We believe that these assumptions can be relaxed to cover the most general Wang-Landau algorithm. Making these four assumptions allows to propose a clearer proof, and we propose hints on how to relax them in Section 4.6.

We denote by U_t the increment of Z_t, such that for any t:

$$Z_{t+1} = Z_t + U_t = Z_t + f(\mathbb{1}_{\mathcal{X}_1}(X_t), \phi_1, \gamma) - f(\mathbb{1}_{\mathcal{X}_2}(X_t), \phi_2, \gamma)$$

Here with only two bins, the increments U_t can take two different values, $+a$ or $-b$, for some $a > 0$ and $b > 0$ that depend on ϕ and γ. For example, with update (4.1) :

$$\begin{cases} a & = 2\gamma(1 - \phi_1) > 0 \\ b & = 2\gamma\phi_1 > 0 \end{cases}$$

whereas with update (4.2) :

$$\begin{cases} a & = \log\left(\frac{1+\gamma(1-\phi_1)}{1-\gamma(1-\phi_1)}\right) > 0 \\ b & = \log\left(\frac{1+\gamma\phi_1}{1-\gamma\phi_1}\right) > 0 \end{cases}$$

and in both cases, if $X_t \in \mathcal{X}_1$ then $U_t = +a$, otherwise $U_t = -b$.

We want to prove that Z_t/t goes to 0, and we are going to prove a stronger result that states, in words, that when Z_t leaves a fixed interval $[\bar{Z}^{lo}, \bar{Z}^{hi}]$, it returns to it in a finite time.

4.3.1 Behaviour of (Z_t) outside an interval

First, lemma 3 states that if Z_t goes above a value $\bar{Z}^{hi}$, it has a strictly positive probability of starting to decrease, and that when that happens, it keeps on decreasing with a high probability.

Lemma 3. *With the introduced processes Z_t and U_t, there exists $\epsilon > 0$ such that for all $\eta > 0$, there exists $\bar{Z}^{hi}$ such that, if $Z_t \geq \bar{Z}^{hi}$, we have the following two inequalities:*

$$P[U_{t+1} = -b | U_t = +a, Z_t] > \epsilon$$
$$P[U_{t+1} = -b | U_t = -b, Z_t] > 1 - \eta.$$

Proof of Lemma 3. We start with the first inequality. Let q_{min} be like in Assumption 3.

In terms of events $\{U_t = +a\}$ is equivalent to $\{X_t \in \mathcal{X}_1\}$, by definition. If $X_t \in \mathcal{X}_1$ and

$\pi(X_t) > 0$ then:

$$\begin{aligned}
K_{\theta_t}(X_t, \mathcal{X}_2) &= \int_{\mathcal{X}_2} K_{\theta_t}(X_t, y) dy \\
&= \int_{\mathcal{X}_2} Q(X_t, y) \rho_{\theta_t}(X_t, y) dy \\
&= \int_{\mathcal{X}_2} Q(X_t, y) \left(1 \wedge \frac{\pi(y)}{\pi(X_t)} \frac{Q(y, X_t)}{Q(X_t, y)} \frac{\theta_t(J(X_t))}{\theta_t(J(y))}\right) dy \\
&= \int_{\mathcal{X}_2} Q(X_t, y) \left(1 \wedge \frac{\pi(y)}{\pi(X_t)} \frac{Q(y, X_t)}{Q(X_t, y)} e^{Z_t}\right) dy
\end{aligned}$$

Using Assumption 4, $\frac{\pi(y)}{\pi(x)} \frac{Q(y,x)}{Q(x,y)}$ is bounded from below, hence there exists K_1 such that:

$$\forall k \geq K_1 \quad \forall x, y \in \mathcal{X} \quad \frac{\pi(y)}{\pi(x)} \frac{Q(y, x)}{Q(x, y)} e^k \geq 1.$$

If $Z_t \geq K_1$ and $X_t \in \mathcal{X}_1$, then:

$$K_{\theta_t}(X_t, \mathcal{X}_2) = \int_{\mathcal{X}_2} Q(X_t, y) dy > q_{min} \mu(\mathcal{X}_2).$$

Hence if $Z_t \geq K_1$:

$$\begin{aligned}
P[U_{t+1} = -b | U_t = +a, Z_t] &= P[X_{t+1} \in \mathcal{X}_2 | X_t \in \mathcal{X}_1, Z_t] \\
&> q_{min} \mu(\mathcal{X}_2).
\end{aligned}$$

We now prove the second inequality. Let us show that for any $\eta > 0$ there exists K_2 such that, provided $Z_t > K_2$:

$$P[U_{t+1} = -b | U_t = -b, Z_t] > 1 - \eta.$$

We have

$$P[U_{t+1} = -b | U_t = -b, Z_t] = P[X_{t+1} \in \mathcal{X}_2 | X_t \in \mathcal{X}_2, Z_t].$$

Again let us first work for a fixed $X_t \in \mathcal{X}_2$.

$$\begin{aligned}
K_{\theta_t}(X_t, \mathcal{X}_2) &= 1 - K_{\theta_t}(X_t, \mathcal{X}_1) \\
&= 1 - \left[\int_{\mathcal{X}_1} Q(X_t, y) \rho_{\theta_t}(X_t, y) dy\right] \\
&= 1 - \left[\int_{\mathcal{X}_1} Q(X_t, y) \left(1 \wedge \frac{\pi(y)}{\pi(X_t)} \frac{Q(y, X_t)}{Q(X_t, y)} e^{-Z_t}\right) dy\right]
\end{aligned}$$

Using the assumption that the MH ratio $\frac{\pi(y)}{\pi(x)} \frac{Q(y,x)}{Q(x,y)}$ is bounded from above, there exists K_2 such that:

$$\forall k \geq K_2 \quad \forall x, y \in \mathcal{X} \quad \frac{\pi(y)}{\pi(x)} \frac{Q(y, x)}{Q(x, y)} e^{-k} \leq 1.$$

And hence for $Z_t > K_2$:

$$\begin{aligned} K_{\theta_t}(X_t, \mathcal{X}_2) &= 1 - e^{-Z_t} \int_{\mathcal{X}_1} Q(X_t, y) \frac{\pi(y)}{\pi(X_t)} \frac{Q(y, X_t)}{Q(X_t, y)} dy \\ &> 1 - e^{-Z_t} \int_{\mathcal{X}_1} Q(X_t, y) M \, dy \\ &> 1 - M e^{-Z_t} \end{aligned}$$

and hence for any η, there is a K_3 greater than K_2 such that for all $Z_t \geq K_3$:

$$K_{\theta_t}(X_t, \mathcal{X}_2) > 1 - \eta$$

We thus obtain:

$$P[U_{t+1} = -b | U_t = -b, Z_t] > 1 - \eta.$$

To conclude we finally define $\epsilon = q_{min} \mu(\mathcal{X}_2)$ and then for any $\eta > 0$, by taking any $\bar{Z}^{hi}$ greater than $K_1 \vee K_3$ we have both inequalities. □

Considering the symmetry of the problem, we instantly have the following corollary result. It states that if Z_t goes too low, it has a strictly positive probability of starting to increase, and when that happens, it keeps on increasing with a high probability.

Lemma 4. *With the introduced processes Z_t and U_t, there exists $\epsilon > 0$ such that for all $\eta > 0$, there exists $\bar{Z}^{lo}$ such that, if $Z_t \leq \bar{Z}^{lo}$, we have the following two inequalities:*

$$\begin{aligned} P[U_{t+1} = +a | U_t = -b, Z_t] &> \epsilon \\ P[U_{t+1} = +a | U_t = +a, Z_t] &> 1 - \eta. \end{aligned}$$

4.3.2 A new process that bounds (Z_t) outside the set

In this section, the proof introduces a new sequence of increments $\tilde{U}_t$ that bounds U_t, and such that the sequence $\tilde{Z}_t$ using $\tilde{U}_t$ as increments:

$$\tilde{Z}_{t+1} = \tilde{Z}_t + \tilde{U}_t$$

returns to $[\bar{Z}^{lo}, \bar{Z}^{hi}]$ in a finite time whenever it leaves it. It will imply that Z_t also returns to $[\bar{Z}^{lo}, \bar{Z}^{hi}]$ in finite time whenever it leaves it. Figure 4.1 might help to visualize the proof.

First let us use Lemma 3. We can take $\epsilon < 1/2$ and $\eta < \min(1/2, \epsilon b/a)$. The Lemma gives the existence of an integer K such that if $Z_t \geq K$, we have the following two inequalities:

$$P[U_{t+1} = -b | U_t = +a, Z_t] > \quad \epsilon \tag{4.4}$$

$$P[U_{t+1} = -b | U_t = -b, Z_t] > \quad 1 - \eta. \tag{4.5}$$

Suppose that there is some time s such that $Z_{s-1} \leq K$ and $Z_s \geq K$. Note that necessarily $Z_s \in [K, K + a]$. Then we define $\tilde{Z}_s = Z_s$, a new process starting at time s. Let $s + T$ be the first time after s such that $Z_{s+T} \leq K$. We wish to show that $E[T] < \infty$.

We define the sequence of random variables $(Z_t)_{t \geq s}$ defined by $\tilde{Z}_s = Z_s$ and $\tilde{Z}_{t+1} = \tilde{Z}_t + \tilde{U}_t$ for $t > s$, where $\left(\tilde{U}_t\right)_{t \geq s}$ is a sequence of random variables taking the values $+a$ or $-b$.

For $s \leq t < T$, $\tilde{U}_t$ is defined as follows:

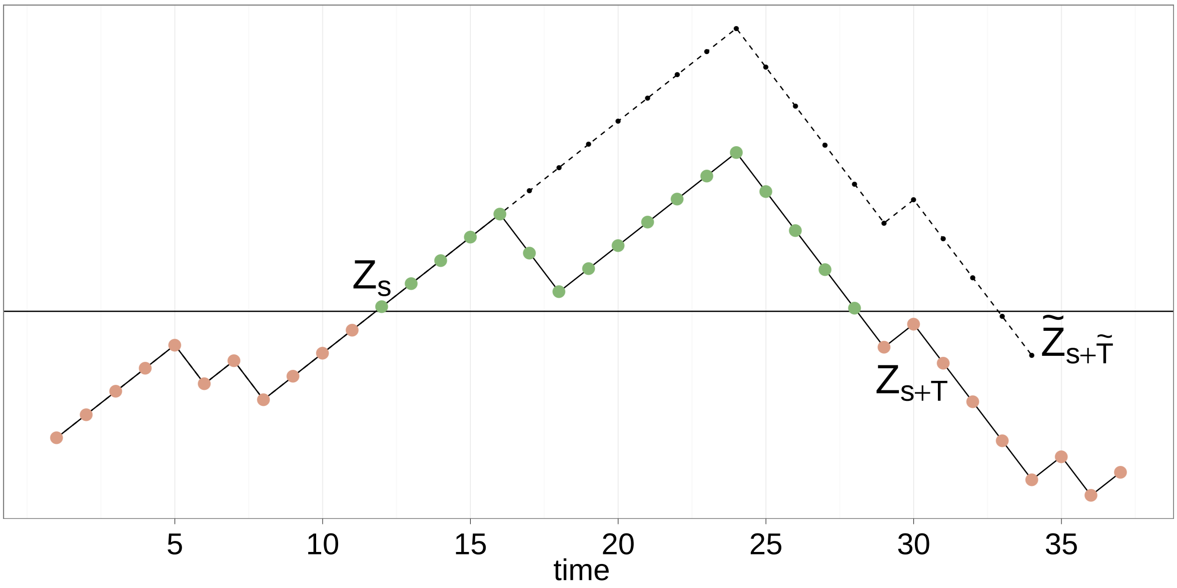


Figure 4.1: Trajectory of Z (full line with dots) and of $\tilde{Z}$ (dotted line), when these processes go above some level $\bar{Z}^{hi}$ indicated by a horizontal full line. Z goes above the level at time s, and returns below it at time $s + T$, whereas $\tilde{Z}$ stays above the level until time $s + \tilde{T}$, with $T \leq \tilde{T}$.

- if $U_{t+1} = +a$ then $\tilde{U}_{t+1} = +a$;
- if $U_{t+1} = -b$, $U_t = -b$ and $\tilde{U}_t = -b$ then $\tilde{U}_{t+1} = -b$ with probability $p_1 = (1-\eta)/P[U_{t+1} = -b|U_t = -b, Z_t]$ and $\tilde{U}_{t+1} = +a$ otherwise;
- if $U_{t+1} = -b$, $U_t = +a$ and $\tilde{U}_t = +a$, then $\tilde{U}_{t+1} = -b$ with probability $p_2 = \epsilon/P[U_{t+1} = -b|U_t = +a, Z_t]$ and $\tilde{U}_{t+1} = +a$ otherwise;
- if $U_{t+1} = -b$, $U_t = -b$ and $\tilde{U}_t = +a$, then $\tilde{U}_{t+1} = -b$ with probability $p_3 = \epsilon(1+P[U_{t+1} = +a|U_t = -b, Z_t]/P[U_{t+1} = -b|U_t = -b, Z_t])$ and $\tilde{U}_{t+1} = +a$ otherwise.

For times $t \geq T$, $\tilde{U}_t$ is a Markov chain independent of U_t and Z_t, with transition matrix

$$\begin{pmatrix} 1-\epsilon & \epsilon \\ \eta & 1-\eta \end{pmatrix}$$

where the first state corresponds to $+a$ and the second state to $-b$.

First, let us check that all these probabilities are indeed less than 1. For p_1, it follows from inequality (4.5). For p_2, it follows from inequality (4.4). For p_3, we have

$$\epsilon\left(1 + \frac{P[U_{t+1} = +a|U_t = -b, Z_t]}{P[U_{t+1} = -b|U_t = -b, Z_t]}\right) \leq \epsilon\left(1 + \frac{\eta}{1-\eta}\right) \leq 2\epsilon \leq 1$$

where we used the conditions $\eta < 1/2$ and $\epsilon < 1/2$. Hence $(\tilde{U}_t)$ is well defined.

Lemma 5. *$(\tilde{U}_t)$ is a Markov chain over the space $\{+a, -b\}$ with transition matrix*

$$\begin{pmatrix} 1-\epsilon & \epsilon \\ \eta & 1-\eta \end{pmatrix}$$

where the first state corresponds to $\{+a\}$ and the second state to $\{-b\}$.

Proof of Lemma 5. We only need to check this for times $t \leq T$. The events $\{\tilde{U}_t = -b\}$ and $\{\tilde{U}_t = -b, U_t = -b\}$ are identical, hence:

$$\begin{aligned} P[\tilde{U}_{t+1} = -b | \tilde{U}_t = -b, Z_t] &= P[\tilde{U}_{t+1} = -b | \tilde{U}_t = -b, U_t = -b, Z_t] \\ &= P[\tilde{U}_{t+1} = -b | U_{t+1} = -b, \tilde{U}_t = -b, U_t = -b, Z_t] \\ &\quad \times P[U_{t+1} = -b | \tilde{U}_t = -b, U_t = -b, Z_t] \\ &= \frac{(1-\eta) P[U_{t+1} = -b | U_t = -b, Z_t]}{P[U_{t+1} = -b | U_t = -b, Z_t]} \\ &= 1 - \eta. \end{aligned}$$

Note that this does not depend on Z_t.

Similarly:

$$\begin{aligned} &P[\tilde{U}_{t+1} = -b | \tilde{U}_t = +a, U_t = +a, Z_t] \\ &\quad = P[\tilde{U}_{t+1} = -b | U_{t+1} = -b, \tilde{U}_t = +a, U_t = +a, Z_t] \\ &\qquad \times P[U_{t+1} = -b | \tilde{U}_t = +a, U_t = +a, Z_t] \\ &\quad = \frac{\epsilon P[U_{t+1} = -b | U_t = +a, Z_t]}{P[U_{t+1} = -b | U_t = +a, Z_t]} \\ &\quad = \epsilon. \end{aligned}$$

And:

$$\begin{aligned} &P[\tilde{U}_{t+1} = -b | \tilde{U}_t = +a, U_t = -b, Z_t] \\ &\quad = P[\tilde{U}_{t+1} = -b | U_{t+1} = -b, \tilde{U}_t = +a, U_t = -b, Z_t] \\ &\qquad \times P[U_{t+1} = -b | \tilde{U}_t = +a, U_t = -b, Z_t] \\ &\quad = \epsilon \left(1 + \frac{P[U_{t+1} = +a | U_t = -b, Z_t]}{P[U_{t+1} = -b | U_t = -b, Z_t]} \right) \\ &\qquad \times P[U_{t+1} = -b | U_t = -b, Z_t] \\ &\quad = \epsilon (P[U_{t+1} = -b | U_t = -b, Z_t] + P[U_{t+1} = +a | U_t = -b, Z_t]) \\ &\quad = \epsilon. \end{aligned}$$

These last two calculations result in:

$$P[\tilde{U}_{t+1} = -b | \tilde{U}_t = +a] = \epsilon$$

with no dependence on Z_t (or U_t). □

The previous lemma is central to the proof, and especially the lack of dependence on Z_t. We always have $\tilde{U}_s = +a$, since $U_s = +a$. Hence for each $t \geq s$, the distribution of $\tilde{U}_t$ depends only on η and ϵ, and implicitly on the threshold K, but not on the value of Z_s. Hence $(\tilde{U}_t)$ has the same law, every time the process (Z_t) goes above K.

4.3.3 Conclusion: proof of Theorem 1 and Corollary 2

Let us now use the bounding process $(\tilde{Z}_t)$ to control the time spent by (Z_t) above K.

Lemma 6. *There exists $\tau \in \mathbb{R}$ such that, for all times s such that $Z_{s-1} \leq K$ and $Z_s \geq K$,* *and defining T by $T = \inf_{d \geq 0}\{Z_{s+d} \leq K\}$, then:*

$$\mathbb{E}[T] \leq \tau$$

Proof of Lemma 6. The Markov chain $(\tilde{U}_t)$ admits the following stationary distribution:

$$\pi_{\tilde{U}} = \left(\frac{\eta}{\epsilon + \eta}, \frac{\epsilon}{\epsilon + \eta}\right)$$

Let us denote by $\tilde{T}$ the time spent by $(\tilde{Z}_t)$ over K, that is:

$$\tilde{T} = \inf_{d \geq 0}\{\sum_{t=s+1}^{s+d} \tilde{U}_t \leq -a\}$$

Remember that $\tilde{Z}_s = Z_s \in [K, K+a]$, hence $\tilde{Z}_{s+\tilde{T}} \leq K$ (whatever the value of Z_s). Now, our choice of η results in $a\eta < b\epsilon$ which implies $E[\tilde{T}] < \infty$ [Norris 98]. Let $\tau = E[\tilde{T}]$. Note that since the law of $(\tilde{U}_t)$ does not depend on the value of Z_s, τ does not depend on Z_s.

Since, for $t \leq T$, we impose that "if $U_{t+1} = +a$ then $\tilde{U}_{t+1} = +a$", it follows that $\forall t \leq T, U_t \leq \tilde{U}_t$. Consequently $\forall t \leq T, Z_t \leq \tilde{Z}_t$ and hence $T \leq \tilde{T}$. Note that (the distribution of) T depends on the exact value of Z_s, but that $\tilde{T}$ as we have defined it has a fixed distribution. We have $E[T] \leq \tau$ (whatever the value Z_s). □

Proof of Theorem 1. Let us define the following sequence of indices:

$$S_1 = \inf_{s \geq 0}\{Z_{s-1} \leq K \text{ and } Z_s \geq K\} \; ; \; S_k = \inf_{s \geq S_{k-1}}\{Z_{s-1} \leq K \text{ and } Z_s \geq K\}$$

The sequence (S_k) represents the times at which the process (Z_t) goes above K. Moreover let us introduce the sequence of time spent above K:

$$T_k = \inf_{s \geq 0}\{Z_{S_k+s-1} \geq K \text{ and } Z_{S_k+s} \leq K\}$$

We have $Z_{S_k} \in [K, K+a]$. Define $k(t)$ such that $S_{k(t)} \leq t < S_{k(t)+1}$. Either $Z_t \leq K$ or $Z_t > K$. In the latter case, $Z_t \leq Z_{S_{k(t)}} + aT_{k(t)}$. Clearly in any case:

$$\mathbb{E}[Z_t] \leq (K+a) + a\tau. \tag{4.6}$$

A similar reasoning on the the lower bound leads to K' and $\tau' < \infty$ such that

$$\mathbb{E}[Z_t] \geq (K'-b) - b\tau'. \tag{4.7}$$

Inequalities (4.7) and (4.6) imply

$$\mathbb{E}\left[\frac{Z_t}{t}\right] \to 0.$$

□

As stated at the beginning of the section, for update (4.1) the convergence $Z_t/t \to 0$ (in mean) implies the convergence of the proportions (ν_t/t) to ϕ (also in mean). We now show that this ensures that the Flat Histogram is reached in finite time.

Proof of Corollary 2. For a fixed threshold c, recall that (FH) being reached at time t

corresponds to the event:

$$\mathrm{FH}_t = \{\forall i \in \{1,\dots,d\} \quad \left|\frac{\nu_t(i)}{t} - \phi_i\right| < c\}$$

We will only use the convergence in probability of the proportions to ϕ for all i:

$$\frac{\nu_t(i)}{t} \xrightarrow[t\to\infty]{\mathbb{P}} \phi_i$$

which implies:

$$\forall \varepsilon > 0 \, \exists N \in \mathbb{N} \, \forall t \geq N \quad \mathbb{P}(FH_t) \geq 1 - \varepsilon$$

We can hence define a stopping time T^{FH} corresponding to the first (FH) being reached:

$$T^{\mathrm{FH}} = \inf_{t\geq 0}\{FH_t\}$$

and some $\varepsilon > 0$ such that:

$$\exists N \in \mathbb{N} \, \forall n \geq N \quad \mathbb{P}(T^{\mathrm{FH}} \leq N + n) \geq \varepsilon$$

Using Lemma 10.11 of [Williams 91], the expectation of T^{FH} is then finite. □

4.4 Proof when $d \geq 2$

In this section we extend the proof to the more general case $d \geq 2$. Having proved that for $d = 2$, only update (4.1) is valid, we now focus on this update and omit update (4.2).

We consider the log penalties defined for update (4.1) by:

$$\log\theta_t(i) = \nu_t(i)(1-\phi_i) - (t - \nu_t(i))\phi_i = \nu_t(i) - t\phi_i$$

where $\nu_t(i)$ is the number of visits of (X_t) in $\mathcal{X}_i$. We assume without loss of generality that $\log\theta_0 = 0$. Then $(X_t, \log\theta_t)$ is a Markov chain, by definition of the WL algorithm. We first prove that $(X_t, \log\theta_t)$ is λ-irreducible, for a sigma-finite measure λ. We will require the following additional assumption on the desired frequencies ϕ.

Assumption 5. *The desired frequencies are rational numbers:*

$$\phi = (\phi_1, \dots, \phi_d) \in \mathbb{Q}^d.$$

Lemma 7. *Let Θ be the following subset of $\mathbb{R}^d$:*

$$\Theta = \{z \in \mathbb{R}^d : \exists (n_1,\dots,n_d) \in \mathbb{N}^d \;\; z_i = n_i - \phi_i S_n \text{ where } S_n = \sum_{j=1}^{d} n_j\}$$

Then denoting by λ the product of the Lebesgue measure μ on $\mathcal{X}$ and of the counting measure on Θ, $(X_t, \log\theta_t)$ is λ-irreducible.

Proof. The proof essentially comes from Bézout's lemma, and is detailed in the Appendix. Note however that it relies on Assumption 5, that was not required for the case $d = 2$. Although not a very satisfying assumption, which is likely not to be necessary for proving the occurrence of (FH) in finite time, it seems to be necessary for the irreducibility of $(X_t, \log\theta_t)$, at least with respect to a standard sigma-finite measure. In any case, this assumption is not restrictive in practice. □

Since this chain is λ-irreducible, the proportion of visits to any λ-measurable set of $\mathcal{X} \times \Theta$ converges to a limit in $[0,1]$. This implies that the vector $(\nu_t(i)/t)$ converges to some vector (p_i). The following is a *reductio ad absurdum*.

Suppose that for some $i \in \{1,\dots,d\}$, $p_i \neq \phi_i$. Since the vectors p and ϕ both sum to 1, this means that for some i, $p_i < \phi_i$: such a state i is visited less than the desired frequency.

Let $\{i_1, i_2, \dots\} = \operatorname{argmin}_{1\le j \le d}(p_j - \phi_j)$. Then for any i_k and for $j \notin \{i_1, i_2, \dots\}$, we have

$$Z_t^{j,i_k} = -\nu_t(i_k) + \nu_t(j) + t(\phi_{i_k} - \phi_j) \sim t(-p_{i_k} + \phi_{i_k} + p_j - \phi_j) \to \infty$$

This implies:

$$\forall K > 0 \, \exists T \in \mathbb{N} \, \forall t > T \quad Z_t^{j,i_k} > K$$

Now consider the stochastic process (U_t) such that

- $U_t = -b$ if $J(X_t) \in \{i_1, i_2, \dots\}$
- $U_t = +a$ otherwise

for some real numbers a and b. Recall that the function J is such that if $X_t \in \mathcal{X}_i$ then $J(X_t) = i$.

Let ϵ be such that when $X_t \notin \mathcal{X}_{i_1} \cup \mathcal{X}_{i_2} \cup \cdots$, there is probability at least ϵ of proposing in $\mathcal{X}_{i_1} \cup \mathcal{X}_{i_2} \cup \cdots$. For large enough K, these proposals will always be accepted. As before, for large enough K, we can make the probability η of leaving $\mathcal{X}_{i_1} \cup \mathcal{X}_{i_2} \cup \cdots$ as small as we wish.

Using the exact same reasoning as in Section 4.3, we can construct a process $(\tilde{U}_t)$ which is a Markov chain with thansition matrix

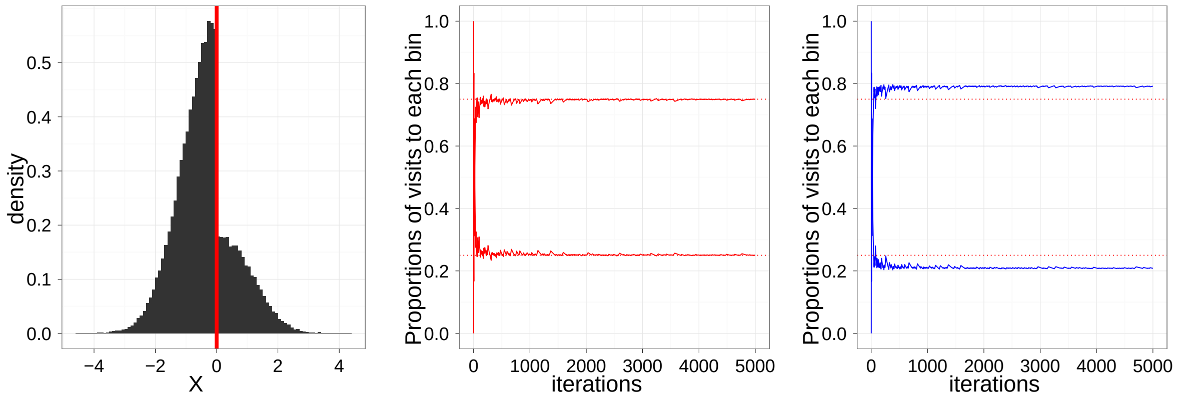
$$\begin{pmatrix} 1-\epsilon & \epsilon \\ \eta & 1-\eta \end{pmatrix}$$

and with $U_t < \tilde{U}_t$ almost surely. Therefore for $t > T$, (U_t) decreases on average, hence (Z_t^{j,i_k}) decreases on average, which contradicts the assumption that it goes to infinity. Hence for all i, $p_i = \phi_i$.

4.5 Illustration of Theorem 1 on a toy example

Let us show the consequences of Theorem 1 on a simple example. We consider as the target distribution the standard normal distribution truncated to the set $\mathcal{X} = [-10, 10]$. We use a Gaussian random walk proposal, with unit standard deviation. Finally we arbitrarily split the state space in $\mathcal{X}_1 = [-10, 0]$ and $\mathcal{X}_2 =]0, 10]$, and we set the desired frequencies to be $\phi = (0.75, 0.25)$. Figure 4.2 shows the results of the Wang-Landau algorithm. Using update (4.1) and 200,000 iterations, we obtain the histogram of Figure 4.2(a). Figure 4.2(b) shows the convergence of the proportions of visits to each bin, using update (4.1). The dotted horizontal lines indicate ϕ, and we can check that the observed proportions of visits converge towards it.

Figure 4.2(c) shows a similar plot, this time using update (4.2). Again, the desired frequencies are represented by dotted lines. Using the left hand side of equation (4.3), we can calculate the theoretical limit of the observed proportion of visits in each bin, which for $\gamma = 1$ and $\phi = (0.75, 0.25)$, is approximately equal to $(0.79, 0.21)$. Hence for a precision threshold c equal to e.g. 1%, the occurrence of (FH) is not likely to occur if one uses update (4.2).



(a) Histogram of the generated sample

(b) Convergence of the proportions of visits to each bin, using the right update

(c) Convergence of the proportions of visits to each bin, using the wrong update

Figure 4.2: Results of the Wang-Landau algorithm using two different updates of the penalties. Histogram of the generated sample using update (4.1), with a vertical line showing the binning (left). Convergence of the proportions of visits to each bin, using update (4.1) (middle) and using update (4.2) (right). The dotted horizontal lines represent the desired frequencies.

As expected, update (4.1) leads to convergence to the desired frequencies but update (4.2) does not.

4.6 Discussion

As seen in Theorem 1 and Corollary 2 of Section 4.3, update (4.1) is valid, in the sense that the frequencies of visits of the chain (X_t) converges towards ϕ. Consequently (FH) is met in finite time, for any threshold $c > 0$.

Regarding the proof of Theorem 1 in the case $d > 2$, we assume that the desired frequencies ϕ are rationals (Assumption 5), which allows to prove that the Markov chain generated by the algorithm (X_t, Z_t) is λ-irreducible for some sigma-finite measure λ. However, our proof requires mainly that the proportions of visits of (X_t) to any bin $\mathcal{X}_i$ converge, which is equivalent to the convergence of (Z_t/t). We believe that results on Random Walks in Random Environments [Zeitouni 06] would allow to remove the rationality assumption.

Assumptions 2–4 could be relaxed by using well-known properties of the Metropolis–Hastings algorithm, from which we did not take advantage here. More precisely, note that the Wang-Landau transition kernel differs from the Metropolis–Hastings only when the proposed points, generated through $Q(\cdot,\cdot)$, land in a different bin than the current position of the chain. Otherwise, the kernel behaves like a Metropolis–Hastings targeting π. Hence under some weaker assumptions than the one we have formulated here, it has recurrence properties.

To conclude, we have shown that for fixed γ, the Flat Histogram criterion is reached in finite time for certain updates. For other updates, the observed frequencies do not converge to the desired frequencies, and so there is a non-zero probability that the Flat Histogram criterion will never be verified. Note that we do not make any claims about the distribution of the sample inside each of the bins $\mathcal{X}_i$ at fixed γ.

Acknowledgements

The authors thank Luke Bornn, Arnaud Doucet, Éric Moulines and Christian P. Robert for helpful comments.

Bibliography

[Andrieu 06] C. Andrieu, E. Moulines & P. Priouret. *Stability of stochastic approximation under verifiable conditions.* SIAM Journal on control and optimization, vol. 44, no. 1, pages 283–312, 2006.

[Andrieu 08] C. Andrieu & J. Thoms. *A tutorial on adaptive MCMC.* Statistics and Computing, vol. 18, no. 4, pages 343–373, 2008.

[Atchadé 09] Y. Atchadé, G. Fort, E. Moulines & P. Priouret. Adaptive Markov chain Monte Carlo: Theory and methods. 2009.

[Atchadé 10] Y. Atchadé & J. Liu. *The Wang-Landau algorithm in general state spaces: applications and convergence analysis.* Statistica Sinica, vol. 20, pages 209–233, 2010.

[Cunha Netto 06] AG Cunha Netto, CJ Silva, AA Caparica & R. Dickman. *Wang-Landau sampling in three-dimensional polymers.* Brazilian journal of physics, vol. 36, no. 3A, pages 619–622, 2006.

[Fort 11] G. Fort, E. Moulines, P. Priouret & P. Vandekerkhove. *A central limit theorem for adaptive and interacting Markov chains.* ArXiv e-prints, July 2011.

[Hastings 70] W. K. Hastings. *Monte Carlo sampling methods using Markov chains and their applications.* Biometrika, vol. 57, no. 1, pages 97–109, 1970.

[Kesten 58] Harry Kesten. *Accelerated Stochastic Approximation.* Annals of Mathematical Statistics, vol. 29, no. 1, pages 41–59, 1958.

[Liang 05] F. Liang. *A Generalized Wang-Landau Algorithm for Monte Carlo Computation.* Journal of the American Statistical Association, vol. 100, no. 472, pages 1311–1327, 2005.

[Malakis 06] A. Malakis, P. Kalozoumis & N. Tyraskis. *Monte Carlo studies of the square Ising model with next-nearest-neighbor interactions.* The European Physical Journal B-Condensed Matter and Complex Systems, vol. 50, no. 1, pages 63–67, 2006.

[Ngo 08] V.T. Ngo & HT Diep. *Phase transition in Heisenberg stacked triangular antiferromagnets: End of a controversy.* Physical Review E, vol. 78, no. 3, page 031119, 2008.

[Norris 98] J.R. Norris. Markov chains. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, UK, 1998.

[Robert 04] C.P. Robert & G. Casella. Monte carlo statistical methods. Springer, 2004.

[Silva 06] CJ Silva, AA Caparica & JA Plascak. *Wang-Landau Monte Carlo simulation of the Blume-Capel model.* Physical Review E, vol. 73, no. 3, page 036702, 2006.

[Tierney 98] L. Tierney. *A note on Metropolis-Hastings kernels for general state spaces.* The Annals of Applied Probability, vol. 8, no. 1, pages 1–9, 1998.

[Wang 01a] F. Wang & DP Landau. *Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram.* Physical Review E, vol. 64, no. 5, page 56101, 2001.

[Wang 01b] F. Wang & DP Landau. *Efficient, multiple-range random walk algorithm to calculate the density of states.* Physical Review Letters, vol. 86, no. 10, pages 2050–2053, 2001.

[Williams 91] David Williams. Probability with martingales. Cambridge University Press, 1991.

[Zeitouni 06] O. Zeitouni. *Random walks in random environments.* Journal of Physics A: Mathematical and General, vol. 39, page R433, 2006.

A Proof of Lemma 6

Let $\Theta \subset \mathbb{R}^d$ be the set of possibly reachable values of the process $(\log \theta_t)$. We define it by:

$$\Theta = \{z \in \mathbb{R}^d : \exists (n_1, \dots, n_d) \in \mathbb{N}^d \;\; z_i = n_i - \phi_i S_n \text{ where } S_n = \sum_{j=1}^{d} n_j\}$$

We want to prove the existence of a measure λ on $\mathcal{X} \times \Theta$ such that the Markov chain $(X_t, \log \theta_t)$ is λ-irreducible. Denote by μ the Lebesgue measure on $\mathcal{X}$ and let $A \in \mathcal{B}(\mathcal{X})$ such that $\mu(A) > 0$, and let $z^\star \in \Theta$. Let us show that for any time s at which $X_s = x_s \in \mathcal{X}$ and $\log \theta_s = z_s \in \Theta$, there exists $t > 0$ such that $X_{s+t} \in A$ and $\log \theta_{s+t} = z^\star$ with strictly positive probability. This will prove the λ-irreducibility of $(X_t, \log \theta_t)$ where λ is the product of the Lebesgue measure μ on $\mathcal{X}$ and the counting measure on Θ.

Note first that for any $n = (n_1, \dots, n_d) \in \mathbb{N}^d$, the process (X_t) can visit exactly n_i times each set $\mathcal{X}_i$ (for all i) between some time $s + 1$ and some time $s + \sum_{i=1}^{d} n_i$, since there is always a non-zero probability of X_{t+1} visiting any $\mathcal{X}_i$ given X_t and $\log \theta_t$ (using Assumptions 3 on the proposal distribution and the form of the MH kernel). More formally, given any $n \in \mathbb{N}^d$ and any time s, denoting $S_n = \sum_{i=1}^{d} n_i$:

$$\mathbb{P}\left(\forall i \in \{1, \dots, d\} \quad \sum_{t=s+1}^{s+S_n} \mathbb{I}_{\mathcal{X}_i}(X_t) = n_i \,\middle|\, X_s, \log \theta_s \right) > 0 \tag{8}$$

Furthermore since $\mu(A) > 0$ and since $(\mathcal{X}_i)_{i=1}^{d}$ is a partition of $\mathcal{X}$ (satisfying Assumption 1 on non-empty bins), there exists $B \subset A$ such that $B \subset \mathcal{X}_{i^\star}$ for some $i^\star \in \{1, \dots, d\}$ and $\mu(B) > 0$. We are going to prove the following statement, which means that there is a "path" between any pair of points in Θ:

Lemma 8.

$$\forall z^1, z^2 \in \Theta \; \exists n \in \mathbb{N}^d \; \forall i \in \{1, \dots, d\} \quad z_i^1 + n_i - \left(\sum_{j=1}^{d} n_j \right) \phi_i = z_i^2$$

Then we will conclude as follows: the Markov chain can go from any (x_s, z_s) to some (x_{s+t-1}, z_{s+t-1}) where z_{s+t-1} can be anywhere in Θ, and then in one final step to (x_{s+t}, z_{s+t}) such that $x_{s+t} \in B$ and $z_{s+t} = z^\star$, since z_{s+t-1} can be chosen such that $z_{s+t} = z^\star$ when $x_{s+t} \in B \subset \mathcal{X}_{i^\star}$.

Proof of Lemma 8. The structure of the proof is the following: we prove that $(\log \theta_t)$ can go from 0 to 0, then from any $z \in \Theta$ to 0, and the possibility of going from 0 to any $z \in \Theta$ comes from the definition of Θ.

Suppose that $\log \theta_0 = (0, \dots, 0)$, and let us prove that the process $(\log \theta_t)$ can go back to 0, ie let us find a vector $n = (n_1, \dots, n_2) \in \mathbb{N}^d$ such that

$$\forall i \in \{1, \dots, d\} \quad 0 = n_i - \phi_i S_n \text{ where } S_n = \sum_{j=1}^{d} n_j$$

Under the rationality assumption on ϕ (Assumtion 5), there exists $(a_i, \dots, a_d) \in \mathbb{N}^d$ and $b \in \mathbb{N}$ such that $\phi_i = a_i / b$ for all i. Now define $n \in \mathbb{N}^d$ as follows:

$$\forall i \in \{1, \dots, d\} \quad n_i = k \prod_{j=1, j \neq i}^{d} \frac{1}{a_j}$$

where $k \in \mathbb{N}$ is such that $n_i \in \mathbb{N}$ for all i. Then, using $\sum_{j=1}^d \phi_j = 1$ one can readily check that:

$$\forall i \in \{1, \ldots, d\} \quad n_i - \phi_i \left(\sum_{j=1}^d n_j \right) = 0$$

Hence the vector n defines a possible path for $(\log \theta_t)$ between 0 and 0, in $S_n = \sum_{j=1}^d n_j$ steps, with a strictly positive probability (using Equation (8)).

A similar reasoning allows to find a possible path from any $z \in \Theta$ to 0. For such a $z \in \Theta$, there exists $(m_1, \ldots, m_d) \in \mathbb{N}^d$ such that

$$\forall i \in \{1, \ldots, d\} \quad z_i = m_i - S_m a_i / b \text{ where } S_m = \sum_{j=1}^d m_j \tag{9}$$

We wish to show that there exits $(k_1, \ldots, k_d) \in \mathbb{N}^d$ such that $k_i - S_k a_i/b = -z_i$ for all i, where $S_k = \sum_{j=1}^d k_j$. To construct $(k_1, \ldots, k_d)$, we use the already introduced vector $(n_1, \ldots, n_d)$ such that $n_i - S_n a_i/b = 0$ for all i, where $S_n = \sum_{j=1}^d n_j$. Putting this together with (9), we get for any $C \in \mathbb{N}$:

$$-z_i + C * 0 = -m_i + C * n_i - \frac{a_i}{b}(CS_n - S_m) \tag{10}$$

For C large enough, for all i, $Cn_i - m_i > 0$. We simply take $k_i = Cn_i - m_i$ for all i. This proves that starting from a point $z \in \Theta$ (by definition reachable from 0), $(\log \theta_t)$ can reach 0 again. □

Chapter 5

Parallel Adaptive Wang–Landau for density exploration

Ce chapitre propose des améliorations méthodologiques à l'algorithme de Wang–Landau, introduit dans le chapitre précédent.

D'une part, N chaînes sont utilisées en parallèle au lieu d'une seule. Ces chaînes intéragissent car elles partagent le même processus de pénalité, utilisé dans l'algorithme pour favoriser les régions de l'espace qui n'ont pas été assez visitées par les chaînes. L'utilisation de N chaînes au lieu d'une seule permet de stabiliser ce processus de pénalité, tout en n'apportant pas de coût supplémentaire si N processeurs parallèles sont disponibles. Par ailleurs, la loi de proposition utilisée dans le noyau de Metropolis–Hastings est adaptative: elle utilise l'échantillon généré pour améliorer les performances de l'algorithme au cours de son exécution, par exemple en visant un certain taux d'acceptation, ou en calibrant sa variance sur la variance empirique de l'échantillon généré. Enfin, la partition de l'espace, nécessaire à la mise en œuvre de la méthode, devient également dynamique: les régions de l'espace peuvent être partagées en plusieurs sous-régions au cours de l'exécution de l'algorithme, afin de faciliter l'exploration de l'espace.

La méthode proposée fait ainsi partie de la classe des algorithmes de Monte Carlo à population de chaînes de Markov. Différents exemples permettent d'illustrer les performances de la méthode et de la comparer à d'autres méthodes: un modèle jouet, un modèle de mélange gaussien sur des données synthétiques, un modèle de sélection de variable appliqué à des données liant des taux de mortalité à des mesures de pollution de l'air, et un modèle d'Ising appliqué à des photographies de la banquise. Les exemples insistent sur la facilité de régler les paramètres algorithmiques, malgré l'hétérogénéité des modèles considérés: petite ou grande dimension de l'espace d'états, espace continu ou discret.

Ce chapitre s'accompagne d'une bibliothèque de fonctions pour le logiciel `R`, appelée `PAWL` pour *Parallel Adaptive Wang–Landau*, qui permet de tester facilement la méthode. La bibliothèque est disponible sur les dépôts officiels de `R` (`Comprehensive R Archive Network`).

Authors
- Luke Bornn (University of British Columbia)
- Pierre E. Jacob (Université Paris-Dauphine, CREST, Paris),
- Arnaud Doucet (University of Oxford)
- Pierre Del Moral (INRIA Bordeaux Sud-Ouest and Université de Bordeaux)

Status To be published in the *Journal of Computational and Graphical Statistics*, arXiv number 1109.3829.

Abstract

While statisticians are well-accustomed to performing exploratory analysis in the modeling stage of an analysis, the notion of conducting preliminary general-purpose exploratory analysis in the Monte Carlo stage (or more generally, the model-fitting stage) of an analysis is an area which we feel deserves much further attention. Towards this aim, this paper proposes a general-purpose algorithm for automatic density exploration. The proposed exploration algorithm combines and expands upon components from various adaptive Markov chain Monte Carlo methods, with the Wang-Landau algorithm at its heart. Additionally, the algorithm is run on interacting parallel chains – a feature which both decreases computational cost as well as stabilizes the algorithm, improving its ability to explore the density. Performance of this new parallel adaptive Wang-Landau (PAWL) algorithm is studied in several applications. Through a Bayesian variable selection example, the authors demonstrate the convergence gains obtained with interacting chains. The ability of the algorithm's adaptive proposal to induce mode-jumping is illustrated through a Bayesian mixture modeling application. Lastly, through a 2D Ising model, the authors demonstrate the ability of the algorithm to overcome the high correlations encountered in spatial models.

Keywords: Markov Chain Monte Carlo, Wang-Landau algorithm, parallel computation.

5.1 Introduction

As improvements in technology introduce measuring devices capable of capturing ever more complex real-world phenomena, the accompanying models used to understand such phenomena grow accordingly. While linear models under the assumption of Gaussian noise were the hallmark of early 20th century statistics, the past several decades have seen an explosion in statistical models which produce complex and high-dimensional density functions for which simple, analytical integration is impossible. This growth was largely fueled by renewed interest in Bayesian statistics accompanying the Markov chain Monte Carlo (MCMC) revolution in the 1990's. With the computational power to explore the posterior distributions arising from Bayesian models, MCMC allowed practitioners to build models of increasing size and nonlinearity.

As a core component of many of the MCMC algorithms discussed later, we briefly recall the Metropolis-Hastings algorithm. With the goal of sampling from a density π, the algorithm generates a Markov chain $(x_t)_{t=1}^T$ with invariant distribution π. From a current state x_t, a new state x' is sampled using a proposal density $q_\eta(x_t, x')$ parametrized by η. The proposed state x' is accepted as the new state x_{t+1} of the chain with probability

$$\min\left(1, \frac{\pi(x')q_\eta(x', x_t)}{\pi(x_t)q_\eta(x_t, x')}\right)$$

and if it is rejected, the new state x_{t+1} is set to the previous state x_t. From this simple algorithmic description, it is straightforward to see that if x_t is in a local mode and the proposal density $q_\eta(x_t, x')$ has not been carefully chosen to propose samples from distant regions, the chain will become stuck in the current mode. This is due to the rejection of samples proposed outside the mode, underscoring the importance of ensuring $q_\eta(x_t, x')$ is intelligently designed.

Though standard MCMC algorithms such as the Metropolis-Hastings algorithm and the Gibbs sampler have been studied thoroughly and the convergence to the target distribution is ensured under weak assumptions, many applications introduce distributions which cannot be sampled easily by these algorithms. Multiple reasons can lead to failure in practice, even if long-run convergence is guaranteed; the question then becomes whether or not the required number of iterations to accurately approximate the density is reasonable given the currently available computational power. Among these reasons, let us cite a few that will be illustrated in later examples: the probability density function might be highly multimodal, in which case the chain can get stuck in local modes. Alternatively or additionally, it might be defined on a high-dimensional state space with strong correlations between the components, in which case the proposal distributions (and in particular their large covariance matrices) are very difficult to tune manually. These issues lead to error and bias in the resulting inference, and may be detected through convergence monitoring techniques (see, e.g., [Robert 04]). However, even when convergence is monitored, it is possible that entire modes of the posterior are missed. To address these issues, we turn to a burgeoning class of Monte Carlo methods which we refer to as "exploratory algorithms."

In the following section, we discuss the traits that allow exploratory MCMC algorithms to perform inference in multimodal, high-dimensional distributions, connecting these traits to existing exploratory algorithms in the process. In Section 3, we detail one of these, the Wang-Landau algorithm, and propose several novel improvements that make it more adaptive, hence easier to use, and also improve convergence. Section 4 applies the proposed algorithm to variable selection, mixture modeling and spatial imaging, before Section 5 concludes.

5.2 Exploratory Algorithms

As emphasized by [Schmidler 11], there are two distinct goals of existing adaptive algorithms. Firstly, algorithms which adapt the proposal according to past samples are largely exploitative, in that they improve sampling of features already seen. However, modes or features not yet seen by the sampler might be quite different from the previously explored region, and as such adaptation might prevent adequate exploration of alternate regions of the state space. As an attempted solution to this problem [Craiu 09] suggest adapting regionally, with parallel chains used to perform the adaptation. Secondly, there exists a set of adaptive algorithms whose goal is to adapt in such a way as to encourage density exploration. These include, for instance, the equi-energy sampler [Kou 06], parallel tempering [Swendsen 86, Geyer 91], and the Wang-Landau [Wang 01a, Wang 01b, Liang 05, Atchadé 10] algorithms among others. The algorithm developed here fits into the latter suite of tools, whose goal is to explore the target density, particularly distant and potentially unknown modes.

Although the aforementioned algorithms have proven efficient for specific challenging inference problems, they are not necessarily designed to be generic, and it is often difficult and time-consuming for practitioners to learn and code these algorithms merely to test a model. As such, while statisticians are accustomed to exploratory data analysis, we believe that there is room for generic exploratory Monte Carlo algorithms to learn the basic features of the distribution or model of interest, particularly the locations of modes and regions of high correlation. These generic algorithms would ideally be able to deal with discrete and continuous state spaces and any associated distribution of interest, and would require as few parameters to tune as possible, such that users can use them before embarking on time-consuming, tailor-made solutions designed to estimate expectations with high precision. In this way one may perform inference and compare between a wide range of models without building custom-purpose Monte Carlo methods for each.

We first describe various ideas that have been used to explore highly-multimodal densities, and then describe recent works aimed at automatically tuning algorithmic parameters of MCMC methods, making them able to handle various situations without requiring much case-specific work from the user.

5.2.1 Ability to Cross Low-Density Barriers

The fundamental problem of density exploration is settling into local modes, with an inability to cross low-density regions to find alternative modes. For densities π which are highly multimodal, or "rugged," one can employ tempering strategies, sampling instead from a distribution proportional to $\pi^{1/\tau}$ with temperature $\tau > 1$. Through tempering, the peaks and valleys of π are smoothed, allowing easier exploration. This is the fundamental idea behind parallel tempering, which employs multiple chains at different temperatures; samples are then swapped between chains, using highly tempered chains to assist in the exploration of the untempered chain [Geyer 91]. [Marinari 92] subsequently proposed simulated tempering, which dynamically moves a single chain up or down the temperature ladder. One may also fit tempering within a sequential Monte Carlo approach, whereby samples are first obtained from a highly tempered distribution; these samples are transitioned through a sequence of distributions converging to π using importance sampling and moves with a Markov kernel [Neal 01, Del Moral 06]. However, using tempering strategies with complex densities, one must be careful of phase transitions, where the density transforms considerably across a given temperature value.

A related class of algorithms works by partitioning the state space along the energy function $-\log \pi(x)$. The idea of slicing, or partitioning, along the energy function is the hallmark of several auxiliary variable sampling methods, which iteratively sample $U \sim \mathcal{U}[0, \pi(X)]$ then $X \sim \mathcal{U}\{X : \pi(X) \geq U\}$. This is the fundamental idea behind the Swendsen–Wang algorithm [Swendsen 87, Edwards 88] and related algorithms (e.g. [Besag 93, Higdon 98, Neal 03]). The equi-energy sampler [Kou 06, Baragatti 12], in contrast to the above auxiliary variable methods, begins by sampling from a highly tempered distribution; once convergence is reached, a new reduced-temperature chain is run with updates from a mixture of Metropolis moves and exchanges of the current state with the value of a previous chain in the same energy band. The process is continued until the temperature reaches 1 and the invariant distribution of the chain is the target of interest. As such, this algorithm works through a sequence of tempered distributions, using previous distributions to create intelligent mode-jumping moves along an equal-energy set.

In a similar vein, the Wang-Landau algorithm [Wang 01a, Wang 01b] also partitions the state space $\mathcal{X}$ along a reaction coordinate $\xi(x)$, typically the energy function: $\xi(x) = -\log \pi(x)$, resulting in a partition $(\mathcal{X}_i)_{i=1}^d$. The algorithm generates a time-inhomogeneous Markov chain that admits an invariant distribution $\tilde{\pi}_t$ at iteration t, instead of the target distribution π itself as e.g. in a standard Metropolis-Hastings algorithm. The distribution $\tilde{\pi}_t$ is designed such that the generated chain equally visits the various regions $\mathcal{X}_i$ as $t \to \infty$. Because the Wang-Landau algorithm lies at the heart of our proposed algorithm, it is extensively described in Section 5.3.

It is worth discussing a similar, recently proposed algorithm which combines Markov chain Monte Carlo and free energy biasing [Chopin 12] and its sequential Monte Carlo counterpart (Chapter 3). The central idea of the latter is to explore a sequence of distributions, successively biasing according to a reaction coordinate ξ in a similar manner. However, we have found the method to be largely dependent on selecting a well-chosen initial distribution π_0, as is usually the case with sequential Monte Carlo methods for static inference. If the initial distribution is not chosen to be flatter than the target distribution, which is possibly the case since the regions of interest with respect to the target distribution are *a priori* unknown, the efficiency of the SMC methods relies mostly on the move steps within the particle filter, which are themselves Metropolis–Hastings or Gibbs moves.

5.2.2 Adaptive Proposal Mechanism

Concurrent with the increasing popularity of exploratory methods, the issue of adaptively fine-tuning MCMC algorithms has also seen considerable growth since the foundational paper of [Haario 01], including a series of conferences dedicated solely to the problem (namely, Adap'ski 1 through 3 among others); see the reviews of [Andrieu 08] and [Atchadé 11] for more details. While the de-facto standard has historically been hand-tuning of MCMC algorithms, this new work finds interest in automated tuning, resulting in a new class of methods called adaptive

MCMC.

The majority of the existing literature focuses on creating intelligent proposal distributions for an MCMC sampler. The principal idea is to exploit past samples to induce better moves across the state space by matching moments of the proposal and past samples, or by encouraging a particular acceptance rate of the sampler. The raison d'être of these algorithms is that tuning MCMC algorithms by hand is both time-consuming and prone to inaccuracies. By automating the selection of the algorithm's parameters, practitioners might save considerable time in their analyses. This feature is pivotal in an automated density exploration algorithm. Due to its exploratory nature, it is likely that the practitioner might not have complete knowledge of even the scale of the density support; as a result, having a proposal distribution which adapts to the density at hand is a crucial step in the automation process.

One must be careful in selecting the type of adaptation mechanism employed to encourage exploration, rather than simply exploiting previously explored modes. For instance, tuning a proposal covariance to a previously visited mode might prevent the algorithm from reaching as yet unexplored modes in the direction of the current mode's minor axis. Additionally, when combined with a progressively biased distribution as in the Wang-Landau algorithm, it is desirable to have a proposal which first samples what it sees well, then later grows in step size to better explore the flattened (biased) distribution.

5.3 Proposed Algorithm

We now develop our proposed algorithm. After recalling the Wang-Landau algorithm, which constitutes the core of our method, we describe three improvements: an adaptive binning strategy to automate the difficult task of partitioning the state space, the use of interacting parallel chains to improve the convergence speed and use of computational resources, and finally the use of adaptive proposal distributions to encourage exploration as well as to reduce the number of algorithmic parameters. We detail at the end of the section how to use the output of the algorithm, which we term parallel adaptive Wang-Landau (PAWL) to answer the statistical problem at hand.

5.3.1 The Wang-Landau Algorithm

As previously mentioned, the Wang-Landau algorithm generates a time-inhomogeneous Markov chain that admits a distribution $\tilde{\pi}_t$ as the invariant distribution at iteration t. The biased distribution $\tilde{\pi}_t$ targeted by the algorithm at iteration t is based on the target distribution π, and modified such that a) the generated chain visits all the sets $(\mathcal{X}_i)_{i=1}^d$ equally, that is the proportion of visits in each set is converging to d^{-1} when t goes to infinity; and b) the restriction of the modified distribution $\tilde{\pi}_t$ to each set $\mathcal{X}_i$ coincides with the restriction of the target distribution π to this set, up to a multiplicative constant. The modification (a) is crucial, as inducing uniform exploration of the sets is the biasing mechanism which improves exploration; in fact similar strategies are used in other fields, including combinatorial optimization [Wei 04]. Ideally the biased distribution $\tilde{\pi}$ would not depend on t, and would be available analytically as:

$$\tilde{\pi}(x) = \pi(x) \times \frac{1}{d} \sum_{i=1}^{d} \frac{\mathcal{I}_{\mathcal{X}_i}(x)}{\psi(i)} \tag{5.1}$$

where $\psi(i) = \int_{\mathcal{X}_i} \pi(x) \mathrm{d}x$ and $\mathcal{I}_{\mathcal{X}_i}(x)$ is equal to 1 if $x \in \mathcal{X}_i$ and 0 otherwise. Checking that using $\tilde{\pi}$ as the invariant distribution of a MCMC algorithm would validate points a) and b) is straightforward. Figure 5.1 illustrates a univariate target distribution π and its corresponding biased distribution $\tilde{\pi}$ under two different partitions of the state space. The biased densities on the middle and right panels have been computed using numerical integration: $\psi(i)$ is approximated for each i (using standard numerical integration, since the toy target density is univariate) and the biased densities are then computed on a fine grid.

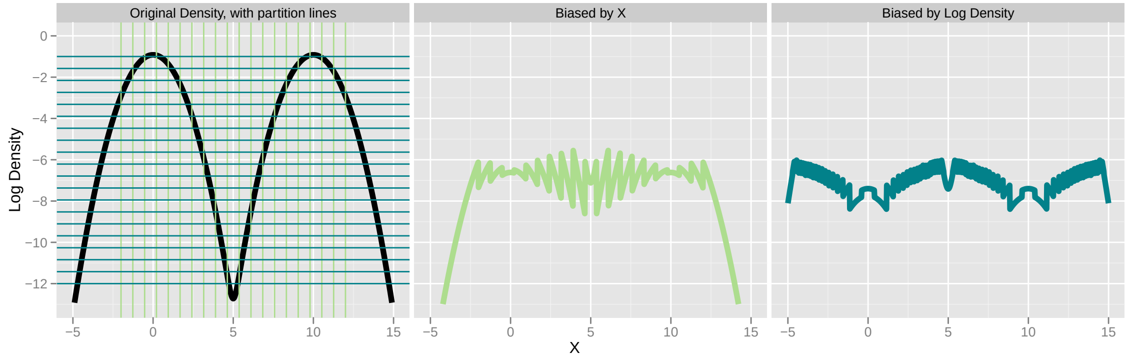


Figure 5.1: Probability density functions for a univariate distribution π and its biased version $\tilde{\pi}$ when partitioning the state space along the x-axis ($\xi(x) = x$, middle) and the log density ($\xi(x) = -\log \pi(x)$, right). The left-most plot also shows the partitioning of the state space with $\xi(x)$; in both cases $d = 20$. The biasing is done such that the integral $\int_{\mathcal{X}_i} \tilde{\pi}(x)\mathrm{d}x$ is the same for all $\mathcal{X}_i$ (areas under the curve for each set) and such that π and $\tilde{\pi}$ coincide on each set $\mathcal{X}_i$, up to a multiplicative constant.

In practical situations, however, the integrals $(\psi(i))_{i=1}^d$ are not available, hence we wish to plug estimates $(\theta(i))_{i=1}^d$ of $(\psi(i))_{i=1}^d$ into Equation (5.1). The Wang-Landau algorithm is an iterative algorithm which jointly generates a sequence of estimates $(\theta_t(i))_t$ for all i and a Markov chain $(X_t)_t$, such that when t goes to infinity, $\theta_t(i)$ converges to $\psi(i)$ and consequently, the distribution of X_t converges to $\tilde{\pi}$. We denote by $\tilde{\pi}_{\theta_t}$ the biased distribution obtained by replacing $\psi(i)$ by its estimate $\theta_t(i)$ in Equation (5.1). Note that the normalizing constant of $\tilde{\pi}_{\theta_t}$ is now unknown. A simplified version of the Wang-Landau algorithm is given in Algorithm 7.

Algorithm 7 Simplified Wang-Landau Algorithm

```
1: Partition the state space into $d$ regions $\{\mathcal{X}_1, \ldots, \mathcal{X}_d\}$ along a reaction coordinate $\xi(x)$.
2: First, $\forall i \in \{1, \ldots, d\}$ set $\theta(i) \leftarrow 1$.
3: Choose a decreasing sequence $\{\gamma_t\}$, typically $\gamma_t = 1/t$.
4: Sample $X_0$ from an initial distribution $\pi_0$.
5: for $t = 1$ to $T$ do
6:     Sample $X_t$ from $P_{\theta_{t-1}}(X_{t-1}, \cdot)$, a transition kernel with invariant distribution $\tilde{\pi}_{\theta_{t-1}}(x)$.
7:     Update the bias: $\log \theta_t(i) \leftarrow \log \theta_{t-1}(i) + \gamma_t(\mathcal{I}_{\mathcal{X}_i}(X_t) - d^{-1})$.
8:     Normalize the bias: $\theta_t(i) \leftarrow \theta_t(i)/\sum_{i=1}^d \theta_t(i)$.
9: end for
```

The rationale behind the update of the bias is that if the chain is in the set $\mathcal{X}_i$, the probability of remaining in $\mathcal{X}_i$ should be reduced compared to the other sets through an increase in the associated bias $\theta_t(i)$. Therefore the chain is pushed towards the sets that have been visited less during the previous iterations, improving the exploration of the state space so long as the partition $(\mathcal{X}_i)_{i=1}^d$ is well chosen. While this biasing mechanism adds cost to each iteration of the algorithm the tradeoff is improved exploration. In step 6, the transition kernel is typically a Metropolis-Hastings move, due to the lack of conjugacy brought about by biasing.

In this simplified form the Wang-Landau algorithm reduces to standard stochastic approximation, where the term γ_t decreases at each iteration. The algorithm as given in [Wang 01a, Wang 01b] uses a more sophisticated learning rate γ_t which does not decrease deterministically, but instead only when a certain criterion is met. This criterion, referred to as the "flat histogram" criterion, is met when for all $i \in \{1, \ldots, d\}$, $\nu(i)$ is close enough to d^{-1}, where we denote by $\nu(i)$ the proportion of visits of (X_t) in the set $\mathcal{X}_i$ since the last time the criterion was met. Hence we introduce a real number c to control the distance between $\nu(i)$ and d^{-1}, and an integer k to count the number of criteria already met. We describe the generalized Wang-Landau in Algorithm 8.

Algorithm 8 Wang-Landau Algorithm

```
Partition the state space into $d$ regions $\{\mathcal{X}_1, \dots, \mathcal{X}_d\}$ along a reaction coordinate $\xi(x)$.
First, $\forall i \in \{1, \dots, d\}$ set $\theta(i) \leftarrow 1, \nu(i) \leftarrow 0$.
Choose a decreasing sequence $\{\gamma_k\}$, typically $\gamma_k = 1/k$.
Sample $X_0$ from an initial distribution $\pi_0$.
for $t = 1$ to $T$ do
    Sample $X_t$ from $P_{\theta_{t-1}}(X_{t-1}, \cdot)$, a transition kernel with invariant distribution $\tilde{\pi}_{\theta_{t-1}}(x)$.
    Update the proportions: $\forall i \in \{1, \dots, d\} \quad \nu(i) \leftarrow \frac{1}{t}\left[(t-1)\nu(i) + \mathcal{I}_{\mathcal{X}_i}(X_t)\right]$.
    if "flat histogram": $\max_{i \in [1,d]} |\nu(i) - d^{-1}| < c/d$ then
        Set $k \leftarrow k + 1$.
        Reset $\forall i \in \{1, \dots, d\} \quad \nu(i) \leftarrow 0$.
    end if
    Update the bias: $\log \theta_t(i) \leftarrow \log \theta_{t-1}(i) + \gamma_k(\mathcal{I}_{\mathcal{X}_i}(X_t) - d^{-1})$.
    Normalize the bias: $\theta_t(i) \leftarrow \theta_t(i) / \sum_{i=1}^{d} \theta_t(i)$.
end for
```

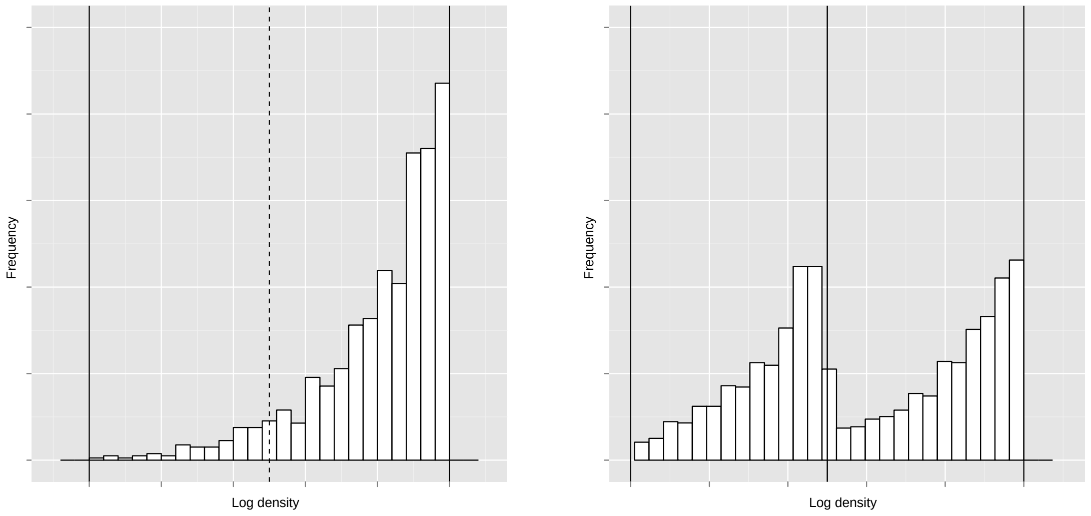
When c is set to low values (e.g. $c = 0.1$ or 0.5), the algorithm must explore the various regions such that the frequency of visits to the region $\mathcal{X}_i$ is approximately d^{-1} before the learning rate γ_k is decreased. Also, the algorithm may be further generalized to target a desired frequency ϕ_i instead of the same frequency d^{-1} for every set; while such strategies may be useful, as demonstrated in the following section, for notational simplicity we focus on the case $\phi_i = d^{-1}$. As already mentioned, to answer the general question of exploring the support of a target density π, the default choice for the reaction coordinate is the energy function: $\xi(x) = -\log \pi(x)$, which has the benefit of being one-dimensional regardless of the dimension of the state space $\mathcal{X}$. However, for specific models other reaction coordinates have been used, such as one (or more) of the components x_j of x or a linear combination of components of x. In the applications in Section 5.4 we discuss the use of alternative reaction coordinates further.

We now propose improvements to the Wang-Landau algorithm to increase its flexibility and efficiency.

5.3.2 A Novel Adaptive Binning Strategy

The Wang-Landau and equi-energy sampler algorithms are known to perform well if the bins, or partitions of the one-dimensional reaction coordinate $\xi(x)$, are well chosen. However, depending on the problem it might be difficult to choose the bins to optimize sampler performance. A typical empirical approach to deal with this issue is to first run, for example, an adaptive MCMC algorithm to find at least one mode of the target distribution. The generated sample and the associated target density evaluations determine a first range of the target density values which can be used to initialize the bins. At this point the user can choose a wider range of target density values (e.g. by multiplying the range by 2), in order to allow for a wider exploration of the space. Within this initial range, one must still decide the number of bins.

Due to difficulties with selecting the bins, it has been suggested that one should adaptively compare adjacent bins, splitting a bin if the corresponding estimate θ is significantly larger than a neighboring value [Schmidler 11]. Because each ψ_i is a given bin's normalizing constant, we feel it is more important to maintain uniformity within a bin to allow easy within-bin movement. Our proposed approach to achieve this "flatness" is to look at the distribution of the realized reaction coordinate values within each bin. Figure 5.2 illustrates this distribution on an artificial histogram. The plot of Figure 5.2(a) shows a situation where, within one bin, the distribution



(a) Unique bin before the split

(b) Two bins after the split

Figure 5.2: Artificial histograms of the log target density values associated to the chains generated by the algorithm, within a single bin (left) and within two bins created by splitting the former bin (right).

might be strongly skewed towards one side. In this artificial example, very few points have visited the left side of the bin, which suggests that moving from this bin to the left neighboring bin might be difficult.

We propose to consider the ratio of the number of points on the left side of the middle (dashed line) over the number of points within the bin as a very simple proxy for the discrepancy of the chains within one bin (see e.g. [Niederreiter 92] for much more sophisticated discrepancy measures). In a broad outline, if this ratio was around 50%, the within-bin histogram would be roughly uniform. On the contrary, the ratio corresponding to Figure 5.2(a) is around 7%. Our strategy is to split the bin if this ratio goes below a given threshold, say 25%; two new bins are created, corresponding to the left side and the right side of the former bin, and each bin is assigned a weight of $\theta/2$ where θ is the weight of the former bin. These provide starting values for the estimation of the weight of the new bins during the following iterations of the algorithm. Note also that the desired frequency of visits to each of the new bins, which was for instance equal to $1/d$ before the split, has to be specified as well. In the numerical experiments, we set the desired frequency of the new bins as one half of the desired frequency of the former bin. Figure 5.2(b) shows the distribution of samples within the two new bins. The resulting histogram is not uniform, yet exhibits a more even distribution within the bin – a feature which is expected to help the chain to move from this bin to the left neighboring bin. The threshold could be set closer to 50%, which would result in more splits and therefore more bins.

In practice it is not necessary to check whether the bins have to be split at every iteration. Our strategy is to check every n-th iteration, until the flat histogram criterion is met for the first time. When it is met, it means that the chains can move easily between the bins, and hence the bins can be kept unchanged for the remaining iterations. Finally, when implementing the automatic binning strategy for discrete distributions, one must ensure that a new bin corresponds to actual points in the state space. For example if the bins are along the energy values and the state space is finite, there are certainly intervals of energy to which no states corresponds, and that would therefore never be reached. Section 4 demonstrates the proposed adaptive binning strategy in practice.

In addition to allowing for splitting of bins, it is also important to allow the range of bins to extend if new states are found outside of the particular range. That said, one must differentiate between the two extremes of the reaction coordinate. For example, if $\xi(x) = -\log \pi(x)$, then one might not wish to add more low-density (high-energy) bins, which would induce the sampler to explore further and further into the tails. However, if one finds a new high-density (low-energy) mode beyond the energy range previously seen, then the sampler might become stuck in this new mode. In this case, we propose to extend the first bin corresponding to the lowest level of energy to always include the lowest observed values. The adaptive partition $(\mathcal{X}_{i,t})_{i=1}^{d_t}$ of the state space takes the following form at time t:

$$\mathcal{X}_{1,t} = [e_{\min,t}, e_{1,t}],\ \mathcal{X}_{2,t} = [e_{1,t}, e_{2,t}],\ \ldots \mathcal{X}_{d_t,t} = [e_{d_t-1,t}, +\infty)$$

where $e_{\min,t} = \min_{t\geq 0}\{-\log \pi(X_t)\}$ and $(e_{1,t}, \ldots, e_{d_t-1,t})$ defines the limits of the inner bins at time t, which is the result of initial bin limits $(e_{1,0}, \ldots, e_{d_0-1,0})$, and possible splits between time 0 and time t. As such, if new low-energy values are found, the bin $\mathcal{X}_{1,t}$ is widened. If this results in unequal exploration across the reaction coordinate, then the adaptive bin-splitting mechanism will automatically split this newly widened bin.

5.3.3 Parallel Interacting Chains

We propose to generate multiple chains instead of a single one to improve computational scalability through parallelization as well as particle diversity. The use of interacting chains has become of much interest in recent years, with the multiple chains used to create diverse proposals [Casarin 11], to induce long-range equi-energy jumps [Kou 06], and to generally improve sampler performance; see [Atchadé 11], [Brockwell 10], and [Byrd 10] for recent developments. The use of parallelization is not constrained to multiple chains, however, and has also been employed to speed up the generation of a single chain through pre-fetching [Brockwell 06].

Let N be the desired number of chains. We follow Algorithm 8, with the following modifications. First we generate N starting points $\boldsymbol{X}_0 = (X_0^{(1)}, \ldots, X_0^{(N)})$ independently from an initial distribution π_0 (Algorithm 8 line 4). Then at iteration t, instead of moving one chain using the transition kernel $P_{\theta_{t-1}}$, we move the N chains using the same transition kernel, associated with the same bias θ_{t-1} (Algorithm 2 line 6). We emphasize that the bias θ is common to all chains, which makes the proposed method different from running Wang-Landau chains entirely in parallel. The proportions $\nu(i)$ are updated using all the chains, simply by replacing the indicator function $\mathcal{I}_{\mathcal{X}_i}(X_t)$ by the mean $N^{-1}\sum_{j=1}^{N}\mathcal{I}_{\mathcal{X}_i}(X_t^{(j)})$, that is the proportion of chains currently in set $\mathcal{X}_i$ (Algorithm 2 line 7). Likewise the update of the bias uses all the chains, again replacing the indicator function by the proportion of chains currently in a given set (Algorithm 2 line 12). We have therefore replaced indicator functions in the Wang-Landau algorithm by the law of the MCMC chain associated with the current parameter. Since this law is not accessible, we perform a mean field approximation at each time step. A similar expression has recently been employed by [Liang 11] for use in parallelizing the stochastic approximation Monte Carlo algorithm [Liang 07, Liang 09]. Note that while we have designed the chains to communicate at each iteration, such frequent message passing can be costly, particularly on graphics processing units. In such situations, one could alter the algorithm such that the chains only communicate periodically.

Our results (see Section 5.4) show that N interacting chains run for T iterations can strongly outperform a single chain run for $N \times T$ iterations, in terms of variance of the resulting estimates. Specifically, having a sample approximately distributed according to $\pi_{\theta_t}(x)$ instead of a single point at iteration t improves and stabilizes the subsequent estimate θ_{t+1}. We explore the tradeoff between N and T in more detail in Section 5.4.

Note that, while the original single-chain Wang-Landau algorithm was not straightforward to parallelize due to its iterative nature, the proposed algorithm can strongly benefit from multiple processing units: at a given iteration the N move steps can be done in parallel, as long as the results are consequently collected to update the bias before the next iteration. Therefore if multiple processors are available, as e.g. in recent central processing units and in graphics processing units (see e.g. [Lee 10, Suchard 09]), the computational cost can be reduced much more than what was possible with the single-chain Wang-Landau algorithm. To summarize, the proposed use of interacting chains can both improve the convergence of the estimates, regardless of the number of available processors, and additionally benefit from multiple processors.

Finally, an additional benefit of using N parallel chains is that they can start from various points, drawn from the initial distribution π_0; hence if π_0 is flat enough, the chains can start from different local modes, which improves *de facto* the exploration. However, we show in Section 5.4 that the chains still explore the space even if they start within the same mode, and hence the efficiency of the method does not rely on the choice of π_0, contrary to what we observed with sequential Monte Carlo methods. Additionally, because the sampler is attempting to explore both the state-space as well as the range of the reaction coordinate simultaneously, our parallel formulation allows the sampler to borrow strength between chains, providing for exploration of the reaction coordinate without having to move a single chain across potentially large and high-dimensional state-spaces to traverse the reaction coordinate values.

5.3.4 Adaptive Proposal Mechanism

As discussed earlier, it is important to automate the proposal mechanism to improve movement across the state space. A well-studied proxy for optimal movement is the algorithm's Metropolis-Hastings acceptance rate. Too low an acceptance rate signifies the algorithm is attempting to make moves that are too large, and are therefore rejected. Too high an acceptance rate signifies the algorithm is barely moving. As such, we suggest adaptively tuning the proposal variance to encourage an acceptance rate of 0.234 as recommended in [Roberts 97], although we have found settings in the range 0.1 to 0.5 to work well in all examples tested. The Robbins-Monro stochastic approximation update of the proposal standard deviation σ_t is as follows:

$$\sigma_{t+1} = \sigma_t + \rho_t \left(2\mathcal{I}(A > 0.234) - 1\right) \tag{5.2}$$

where t is the current iteration of the algorithm, ρ_t is a decreasing sequence (typically $\rho_t = 1/t$), and A is the acceptance rate (proportion of accepted moves) of the particles. Through this update, the proposal variance grows after samples are accepted, and shrinks when samples are rejected, encouraging exploration of the state space.

Another approach to adaptively tuning the proposal distribution is to use the following mixture of Gaussian random-walks:

$$X^* \sim w_1 \mathcal{N}\left(X_{t-1}, \frac{(2.38)^2}{p}\Sigma_t\right) + w_2 \mathcal{N}\left(X_{t-1}, \frac{(\sigma_I)^2}{p} I_p\right) \tag{5.3}$$

with $w_1 + w_2 = 1$, Σ_t the empirical covariance of the chain history – an estimator of the covariance structure of the target – and I_p the $p \times p$ identity matrix where p is the dimension of the target space. The first component of this mixture makes the proposal adaptive and able to learn from the past, while the second component helps to explore the space. For instance, if the chain is stuck in a mode, the first component's variance might become small, yet the second component guarantees a chance to eventually escape the mode. Hence the second component acts as a "safety net" and therefore its weight is small, typically $w_2 = 0.05$, and its standard deviation σ_I may be set large to improve mixing [Guan 07].

In our context where parallel chains are run together, we use all the chains to estimate the empirical covariance Σ_t at each iteration. Note that the computation of this covariance does not require the storage of the whole history of the chain and can be done at constant cost, since recurrence formulae exist to compute the empirical covariance, as explained for instance in [Welford 62]. The value 2.38^2 is justified by asymptotic optimality reflections on certain classes of models (see, e.g., [Roberts 97] and [Roberts 09]).

5.3.5 Using the Output to Perform Inference

While the resulting samples from the proposed algorithm PAWL are not from π, but rather an approximation of the biased version (5.1), one can use importance sampling or advanced sequential

Monte Carlo ideas to transition the samples to π (see Chapter 3 for details). Alternatively, the samples from the exploratory algorithm can be used to seed a more traditional MCMC algorithm, as advocated by [Schmidler 11].

The pseudo-code for PAWL, combining the parallel Wang-Landau algorithm with adaptive binning and proposal mechanisms, is given in the Appendix. Before proceeding to examples, it is important to reiterate the importance of the values $\psi(i) = \int_{\mathcal{X}_i} \pi(x)\mathrm{d}x$. Specifically, certain choices of the reaction coordinate $\xi(x)$ result in $\psi(i)$ having inherent value. For example, it is possible in a model selection application to use the model order as $\xi(x)$, in which case the values $\psi(i)$ could be employed to calculate Bayes factors and other quantities of interest.

5.4 Applications

We now demonstrate PAWL applied to three examples including variable selection, mixture modeling, and spatial imaging. A fourth pedagogical example is available in the appendices. In each application we walk through our proposed algorithm (described explicitly as Algorithm 9 in the Appendix), first running preliminary (adaptive) Metropolis-Hastings MCMC to determine the initial range for the reaction coordinate ξ and initial values for the proposal parameters and starting state of the interacting Wang-Landau chains. This range is then increased to encourage exploration of low-density regions of the space, and an initial number of bins is specified. Once this groundwork is set, the same Metropolis-Hastings algorithm run in the preliminary stage is embedded within the PAWL algorithm.

5.4.1 g-Prior Variable Selection

We proceed by conducting variable selection on the pollution data set of [McDonald 73], wherein mortality is related to pollution levels through 15 independent variables including mean annual precipitation, population per household, and average annual relative humidity. Measured across 60 metropolitan areas, the response variable $\mathbf{y}$ is the age-adjusted mortality rate in the given metropolitan area. Our goal is to identify the pollution-related independent variables which best predict the response. With 15 variables, calculating the posterior probabilities of the 32,768 models exactly is possible but time-consuming. We have chosen this size of data set to provide for difficult posterior exploration, yet allow a study of convergence of θ towards ψ.

With an eye towards model selection, we introduce the binary indicator variable $\gamma \in \{0,1\}^p$, where $\gamma_j = 1$ means the variable $\boldsymbol{x}_j$ is included in the model. As such, γ can describe all of the 2^p possible models. Consider the normal likelihood

$$\mathbf{y}|\mu, \mathbf{X}, \boldsymbol{\beta}, \sigma^2 \sim N_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n). \tag{5}$$

If $\mathbf{X}_\gamma$ is the model matrix which excludes all $\boldsymbol{x}_j$'s if $\gamma_j = 0$, we can employ the following prior distributions for $\boldsymbol{\beta}$ and σ^2 [Zellner 86, Marin 07]:

$$\pi(\boldsymbol{\beta}_\gamma, \sigma^2|\gamma) \propto (\sigma^2)^{-(q_\gamma+1)/2-1} \exp\left[-\frac{1}{2g\sigma^2}\boldsymbol{\beta}_\gamma^T(\mathbf{X}_\gamma^T\mathbf{X}_\gamma)\boldsymbol{\beta}_\gamma\right].$$

where $q_\gamma = \mathbf{1}_n^T\gamma$ represents the number of variables in the model. While selecting g can be a difficult problem, we have chosen it to be very large ($g = \exp(20)$) to induce a sparse model, which is difficult to explore due to the small marginal probabilities of most variables. After integrating over the regression coefficients $\boldsymbol{\beta}$, the posterior density for γ is thus

$$\pi(\gamma|\mathbf{y}, \mathbf{X}) \propto (g+1)^{-(q_\gamma+1)/2}\left[\mathbf{y}^T\mathbf{y} - \frac{g}{g+1}\mathbf{y}^T\mathbf{X}_\gamma(\mathbf{X}_\gamma^T\mathbf{X}_\gamma)^{-1}\mathbf{X}_\gamma\mathbf{y}\right]^{-n/2}. \tag{6}$$

While we select the log energy function $-\log\pi(x)$ as the reaction coordinate $\xi(x)$ for our analysis, it is worth noting that many other options exist. For instance, it would be natural to

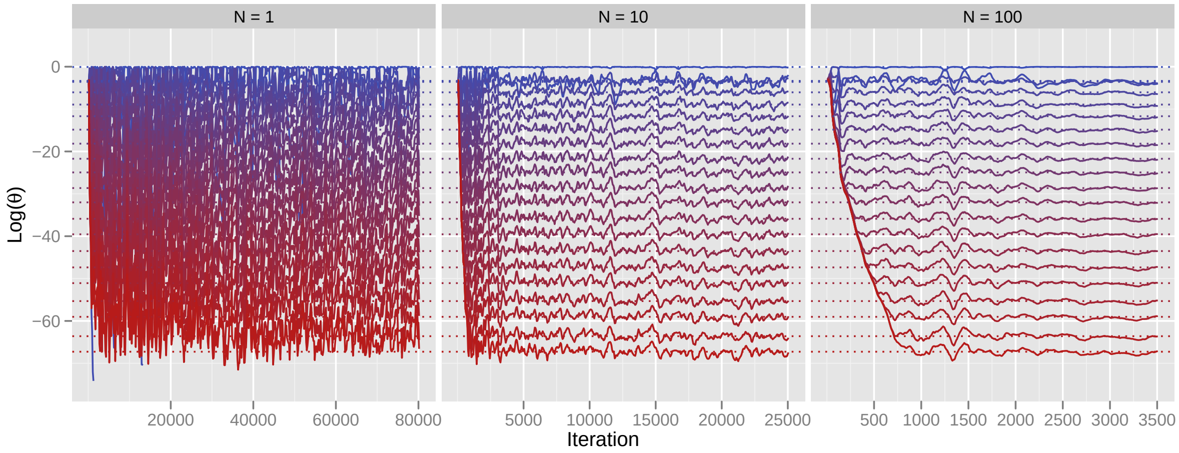


Figure 5.3: Variable selection example: convergence of Wang-Landau for $N = 1, 10, 100$. Iterations set such that each algorithm runs in 2 minutes (±5 seconds). θ for each bin shown in solid lines. True values (ψ) shown as dotted lines.

consider the model saturation q_γ/p, which would ensure exploration across the different model sizes. However, we select $\xi(x) = -\log\pi(x)$ to emphasize the universality of using the energy function as the reaction coordinate.

We first run a preliminary Metropolis-Hastings algorithm which flips a variable on/off at random, accepting or rejecting the flip based on the resulting posterior densities. Due to high correlation between variables, a better strategy might be to flip multiple variables at once; however, we restrain from exploring this to demonstrate PAWL's ability to make viable even poorly designed Metropolis-Hastings algorithms. The preliminary algorithm run found values $377 < -\log\pi(x) < 410$, which we extend slightly to create 20 equally spaced bins in the range $[377, 450]$. It is worth reiterating that the resulting samples generated from PAWL are from a biased version of π; as such, importance sampling techniques could be used to recover π, or the samples obtained could be used to seed a more traditional MCMC algorithm.

Due to the size of the problem, we are able to enumerate all posterior values, and hence may calculate ψ exactly. As such, we begin by examining the effect of the number of particles N on the parallel Wang-Landau algorithm. To further focus on this aspect, we suppress adaptive binning and proposals for this example. Figure 5.3 shows the convergence of θ to Ψ for $N = 1, 10, 100$. We see that the algorithm's convergence improves with more particles. Using $N = 100$ particles, we now examine PAWL compared to the Metropolis-Hastings algorithm (run on N chains) mentioned above on the unnormalized targets $\pi, \pi^{1/10}, \pi^{1/100}$. Consider Figure 5.4; on the target distribution π, the Metropolis-Hastings algorithm becomes stuck in high-probability regions. However, on the tempered distributions, the algorithm explores the space more thoroughly, although not to the same level as PAWL. Specifically, PAWL explores a much wider range of models, including the highest probability models, whereas the tempered distributions do not. Here the Wang-Landau algorithm as well as the Metropolis-Hastings algorithm both use $N = 100$ chains for $T = 3500$ iterations, the former taking 253 ± 13 seconds and the latter taking 247 ± 15 seconds across 10 runs, indicating that the additional cost for PAWL is negligible.

5.4.2 Mixture Modeling

Mixture models provide a challenging case of multimodality, due partly to the complexity of the model and partly to a phenomenon called "label switching" (see e.g. [Frühwirth-Schnatter 06] for a book covering Bayesian inference for these models, [Diebolt 94] and [Richardson 97] for seminal papers using MCMC for mixture models, and [Stephens 00] and [Jasra 05] on the label switching problem). Articles describing explorative MCMC algorithms often take these models

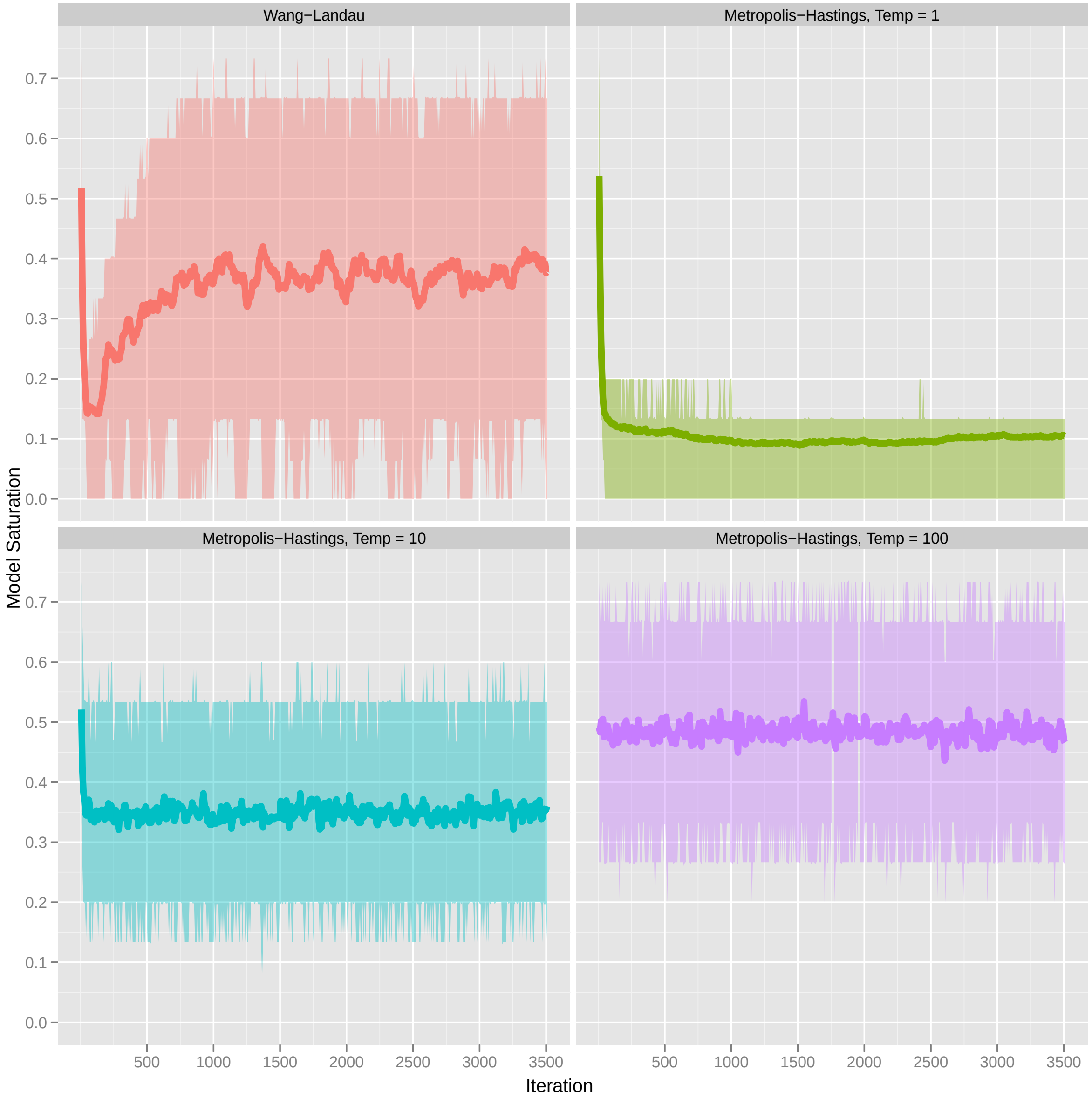


Figure 5.4: Variable selection example: exploration of model space by PAWL and Metropolis-Hastings at 3 temperature settings. (a) Model Saturation (proportion of non-zero variables in model) as function of algorithm iterations. The solid lines are the mean of $N = 100$ chains, while the shaded regions are the middle 95% of the chains.

as benchmarks, as e.g. population MCMC and SMC methods in [Jasra 07], the Wang-Landau algorithm in [Atchadé 10], free energy methods in [Chopin 12] and Chapter 3, and parallel tempering with equi-energy moves in [Baragatti 12].

Consider a Bayesian Gaussian mixture model, i.e. for $i = 1, \dots n$,

$$p(y_i|q,\mu,\lambda) = \sum_{k=1}^{K} q_k\, \varphi(y_i;\mu_k,\lambda_k^{-1}),$$

where φ is the probability density function of the Gaussian distribution, K is the number of components, q_k, μ_k and λ_k are respectively the weight, the mean and the precision of component k. The component index k is also called its label. Following [Richardson 97], the prior is taken as, for $k = 1, \dots, K$,

$$\begin{aligned} \mu_k &\sim \mathrm{N}(M,\kappa^{-1}), & \lambda_k &\sim \mathrm{Gamma}(\alpha,\beta), \\ \beta &\sim \mathrm{Gamma}(g,h), & (q_1,\dots,q_{K-1}) &\sim \mathrm{Dirichlet}_K(1,\dots,1) \end{aligned}$$

with, e.g., $\kappa = 4/R^2$, $\alpha = 2$, $g = 0.2$, $h = 100g/\alpha R^2$, $M = \bar{y}$, R =range(y).

The invariance of the likelihood to permutations in the labelling of the components leads to the "label switching" problem: since there are $K!$ possible permutations of the labels, each mode has necessarily $K! - 1$ replicates. We emphasize that this model has been thoroughly studied and is hence well-understood from a modeling point of view, but it still induces a computationally challenging sampling problem for which difficulty can be artificially increased through the number of components K.

Note that in this parametrization β, the rate of the Gamma prior distribution of the precisions λ_k, is estimated along with the parameters of interest $q_{1:K-1}$, $\mu_{1:K}$ and $\lambda_{1:K}$. [Chopin 12] (and also Chapter 3) suggest that β can be used as a reaction coordinate, since a large value of β results in a small precision and hence in a flatter posterior distribution of the other parameters, which is easier to explore than the distribution associated with smaller values of β; we refer to these articles for further exploration of this choice of reaction coordinate, and instead default to $\xi(x) = -\log \pi(x)$.

We create a synthetic 100-sample from a Gaussian mixture with $k = 4$ components, weights $1/4$, means -3, 0, 3, 6 and variances 0.55^2 as in [Jasra 05]. The goal is to explore the highly multimodal posterior distribution of the 13-dimensional parameter $\theta = (w_{1:4}, \mu_{1:4}, \lambda_{1:4}, \beta)$ where w_k is the unnormalized weight: $q_k = w_k / \sum_{k=1}^{K} w_k$. Unnormalized weights may be handled straightforwardly in MCMC algorithms since they are defined on $\mathbb{R}^+$ and not on the K-simplex as with the q_k.

The proposed algorithm is compared to a Sequential Monte Carlo sampler (SMC) and a parallel adaptive Metropolis–Hastings (PAMH) algorithm, that we detail below. We admittedly use naive versions of these competitors, arguing that most improvements of these could be carried over to PAWL. For instance, a mixture of Markov kernels as suggested for the SMC algorithm in Section 3.2 of [Jasra 07] can be used in the proposal distribution of PAWL; and since PAWL is a population MCMC algorithm, exchange and crossover moves could be used as well, as suggested for the Population SAMC algorithm in [Liang 11]. To get a plausible range of values for the reaction coordinate of the proposed algorithm without user input, an initial adaptive MCMC algorithm is run with $N = 10$ chains and $T^{\text{init}} = 1,000$ iterations. The initial points of these chains are drawn from the prior distribution of the parameters. This provides a range of log density values, from which we compute the 10% and 90% empirical quantiles, denoted by q_{10} and q_{90} respectively. In a conservative spirit, the bins are chosen to equally divide the interval $[q_{10},\ q_{10} + 2(q_{90} - q_{10})]$ in 20 subsets. Hence the algorithm is going to explore log density values in a range that is approximately twice as large as the values initially explored. Note that we use quantiles instead of minimum and maximum values to make the method more robust.

Next, PAWL itself is run for $T = 200,000$ iterations, starting from the terminal points of the N preliminary chains, resulting in a total number of $N(T + T_{\text{init}})$ target density evaluations. In this situation, even with only 100 data points, most of the computational cost goes into the evaluation of the target density. This confirms that algorithmic parameters such as the number of bins does not significantly affect the overall computational cost, at least as long as the target density is not extremely cheap to evaluate. The adaptive proposal is such that it targets an acceptance rate of 23.4%. Meanwhile the PAMH algorithm using the same adaptive proposal is run with $N = 10$ chains and $T^\star = 250,000$ iterations, hence relying on more target density evaluations for a comparable computational cost.

Finally, the SMC algorithm is run on a sequence of tempered distribution $(\pi_k)_{k=1}^K$, each density being defined by:

$$\pi_k(x) \propto \pi^{\zeta_k}(x) p_0^{1-\zeta_k}(x)$$

where p_0 is an initial distribution (here taken to be the prior distribution), and $\zeta_k = k/K$. The number of steps K is set to 100 and the number of particles to 40,000. When the Effective Sample Size (ESS) goes below 90%, we perform a systematic resampling and 5 consecutive Metropolis–Hastings moves. We use a random walk proposal distribution, which variance is taken to be $c\hat{\Sigma}$ where $\hat{\Sigma}$ is the empirical covariance of the particles and c is set to 10%; see [Jasra 07] for more details. The parameters are chosen to induce a computational cost comparable to the other methods. However for the SMC sampler the number of target density evaluations is a random number, since it depends on the random number of resampling steps: the computational cost is in general less predictable than using MCMC.

First we look at graphical representations of the generated samples. Figure 5.5 shows the resulting points projected on the (μ_1, μ_2) plane, restricted on $[-5, 9]^2$. In this plane there are 12 replicates of each mode, indicated by target symbols in Figure 5.5(a). These projections do not allow one to check that all the modes were visited since they project the 13-dimensional target space on a 2-dimensional space. Figure 5.5(b) shows that the adaptive MCMC method clearly misses some of the modes, while visiting many others. Figure 5.5(c) shows how the chains generated by the modified Wang-Landau algorithm easily explore the space of interest, visiting both global and local modes. To recover the main modes from these chains, we use the final value of the bias, θ_T, as importance weights to correct for the bias induced by the algorithm; in Figure 5.5(c) the importance weights define the transparency of each point: the darker the point, the more weight it has. Finally Figure 5.5(d) shows how the SMC sampler also put particles in each mode; again the transparency of the points is proportional to their weights.

We now turn to more quantitative measures of the error made on marginal quantities. Since the component means admit 4 identical modes around -3, 0, 3 and 6, we know that their overall mean is approximately equal to $\mu^\star = 1.5$. We then compute the following error measurement:

$$\text{error} = \sqrt{\sum_{k=1}^{K} (\hat{\mu}_k - \mu^\star)^2}$$

where $\hat{\mu}_k$ is the mean of the generated sample (taking the weights into account for PAWL and SMC). Table 5.1 shows the results averaged over 10 independent runs: the means of each component, the error defined above and the (wall-clock) run times obtained using the same CPU, along with their standard deviations. The results highlight that in this context PAWL gives more precise results than SMC, for the same or less computational cost; the comparison between parallel MCMC and SMC confirms the results obtained in [Jasra 07]. The small benefit of PAWL over PAMH can be explained by considering the symmetry of the posterior distribution: even if some modes are missed by PAMH as shown in Figure 5.5(b), the approximation of the posterior expectation might be accurate, though the corresponding variance will be higher.

Next we consider a more realistic setting, where the initial distribution is not well spread over the parameter values: instead of taking the prior distribution itself, we use a similar distribution but with an hyperparameter κ equal to 1 instead of $4/R^2$, which for our simulated data set is equal to 0.03. This higher precision makes the initial distribution concentrated on a few modes, instead of being fairly flat over the whole region of interest. We keep the prior unchanged, so that the posterior is left unchanged. For PAWL and PAMH, this means that the initial points

Method	μ_1	μ_2	μ_3	μ_4	Error	Time (s)
PAWL	1.42 ± 0.99	1.42 ± 0.58	1.39 ± 0.90	1.75 ± 0.78	1.50 ± 0.59	209 ± 1
PAMH	1.58 ± 0.81	1.25 ± 0.72	1.04 ± 1.07	2.09 ± 1.00	1.75 ± 0.80	233 ± 1
SMC	1.00 ± 1.96	2.99 ± 1.38	0.92 ± 2.27	1.10 ± 2.11	3.89 ± 1.34	269 ± 7

Table 5.1: Estimation of the means of the mixture components, for the proposed method (PAWL), Parallel Adaptive Metropolis–Hastings (PAMH) and Sequential Monte Carlo (SMC), using the prior as initial distribution. Quantities averaged over 10 independent runs for each method.

of the chains are all close one to another; and likewise for the initial particles in the SMC sampler. The results are shown in Table 5.2, and illustrate the degeneracy of SMC when the initial distribution is not well-chosen; though this is not surprising, this is important in terms of exploratory algorithms when one does not have prior knowledge of the region of interest. Both parallel MCMC methods give similar results as with the previous, flatter initial distribution.

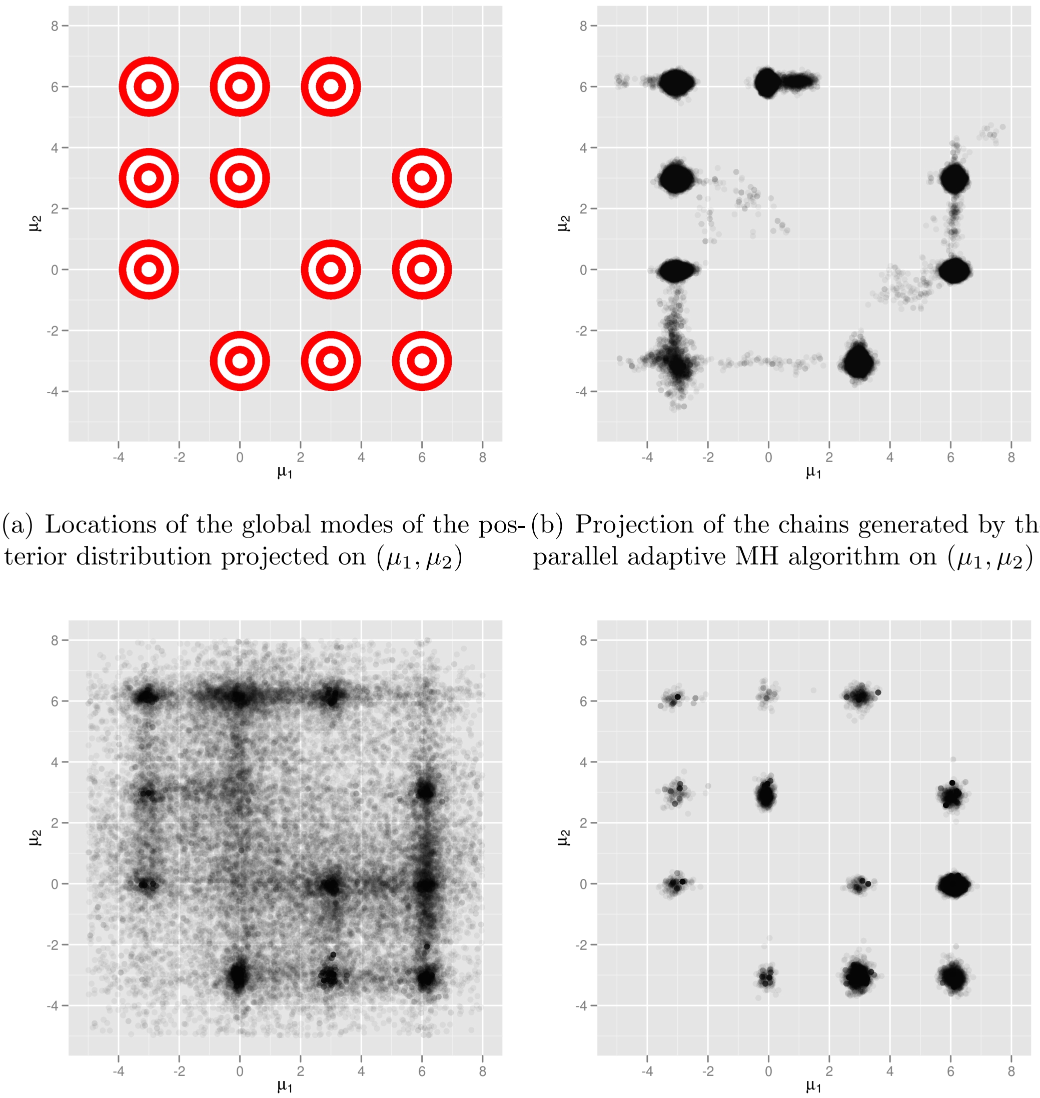
Method	μ_1	μ_2	μ_3	μ_4	Error	Time
PAWL	1.16 ± 0.75	2.04 ± 0.50	1.72 ± 0.80	1.07 ± 1.22	1.48 ± 1.10	210 ± 1
PAMH	1.37 ± 0.73	1.48 ± 1.39	1.71 ± 0.81	1.44 ± 1.11	1.75 ± 1.01	234 ± 1
SMC	0.35 ± 2.13	0.82 ± 1.55	3.19 ± 2.41	1.62 ± 1.85	4.17 ± 1.41	337 ± 8

Table 5.2: Estimation of the means of the mixture components, for the proposed method (PAWL), Parallel Adaptive Metropolis–Hastings (PAMH) and Sequential Monte Carlo (SMC), using a concentrated initial distribution. Quantities averaged over 10 independent runs for each method.

Finally, we compare different algorithmic settings for the PAWL algorithm, changing the number of chains and the number of iterations. The results are shown in Table 5.3. First we see that, even on a single CPU, the computing time is not exactly proportional to $N \times T$, the number of target density evaluation. Indeed the computations are vectorized by iteration, and hence it is typically cheaper to compute one iteration of N chains than N iterations of 1 chain; although this would not hold for every model. We also see that the algorithm using only one chain failed to explore the modes, resulting in a huge final error. Finally we see that with 50 chains and only $50,000$ iterations, the algorithm provides results of approximately the same precision as with 10 chains and $200,000$ iterations. This suggests that the algorithm might be particularly interesting if parallel processing units are available, since the computational cost would then be much reduced.

Parameters	μ_1	μ_2	μ_3	μ_4	Error	Time
$N = 1$ $T = 5 \times 10^5$	0.37 ± 3.46	2.01 ± 3.27	2.53 ± 3.04	0.95 ± 3.46	6.39 ± 1.30	265 ± 40
$N = 10$ $T = 2 \times 10^5$	1.42 ± 0.99	1.42 ± 0.58	1.39 ± 0.90	1.75 ± 0.78	1.50 ± 0.59	209 ± 1
$N = 50$ $T = 5 \times 10^4$	1.51 ± 0.88	1.5 ± 0.9	1.65 ± 0.64	1.31 ± 0.31	1.22 ± 0.69	178 ± 2

Table 5.3: Estimation of the means of the mixture components, for the proposed method (PAWL), for different values of N, the number of chains, and T, the number of iterations. Quantities averaged over 10 independent runs for each set of parameters.



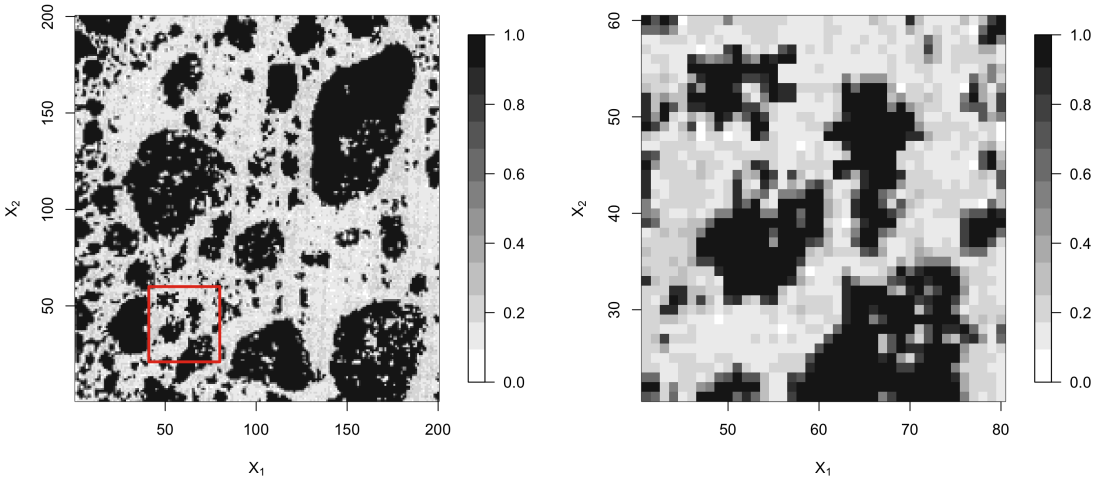
(a) Locations of the global modes of the posterior distribution projected on (μ_1, μ_2)

(b) Projection of the chains generated by the parallel adaptive MH algorithm on (μ_1, μ_2)

(c) Projection of the chains generated by PAWL on (μ_1, μ_2)

(d) Projection of the particles generated by the SMC algorithm on (μ_1, μ_2)

Figure 5.5: Mixture model example: exploration of the posterior distribution projected on the μ_1, μ_2 plane, using different algorithms.



(a) Original Image

(b) Focused Region of Image

Figure 5.6: Spatial model example: (a) original ice floe image with highlighted region. (b) close-up of focused region.

5.4.3 Spatial Imaging

We finish our examples by identifying ice floes from polar satellite images as described in [Banfield 92]. Here the image under consideration is a 200 by 200 gray-scale satellite image, with focus on a particular 40 by 40 region (y, Figure 5.6); the goal is to identify the presence and position of polar ice floes (x). Towards this goal, [Higdon 98] employs a Bayesian model. Basing the likelihood on similarity to the image and employing an Ising model prior, the resulting posterior distribution is

$$\log(\pi(x|y)) \propto \alpha \sum_i I[y_i = x_i] + \beta \sum_{i \sim j} I[x_i = x_j].$$

The first term, the likelihood, encourages states x which are similar to the original image y. The second term, the prior, favors states x for which neighbouring pixels are equal. Here neighbourhood ($\sim$) is defined as the 8 vertical, horizontal, and diagonal adjacencies of each (interior) pixel.

Because the prior strongly prefers large blocks of identical pixels, an MCMC method which proposes to flip one pixel at a time will fail to explore the posterior, and hence [Higdon 98] suggests a partial decoupling technique specific to these types of models. However, to demonstrate PAWL's power and universality, we demonstrate its ability to make simple one-at-a-time Metropolis-Hastings feasible in these models without more advanced decoupling methods.

First running a preliminary Metropolis-Hastings algorithm of length $20,000$, we use the range of explored energy values divided evenly across 10 bins. The algorithm subsequently splits bins 6 times (with splitting stopped once the algorithm reaches the extremes of the reaction coordinate values) resulting in 17 bins at the algorithm's conclusion. For both algorithms, we run 10 chains for $1,000,000$ iterations with model parameters $\alpha = 1$, $\beta = 0.7$. Due to the flip-one-pixel approach, we suppress adaptive proposals for this example. In contrast to the mixture modeling example, in this example the target density is fairly straightforward to calculate, so it is a good worst-case comparison to demonstrate the additional time taken by the proposed algorithm. For this example, the MH algorithm took 388 ± 21 seconds across 10 runs, whereas PAWL required 478 ± 24 seconds. Thus in this case the Wang-Landau adds a 23% price to each iteration on average. However, as we will show, the exploration is significantly better, justifying the slight additional cost. Figure 5.7 shows a subset of the last $400,000$ posterior realizations from one

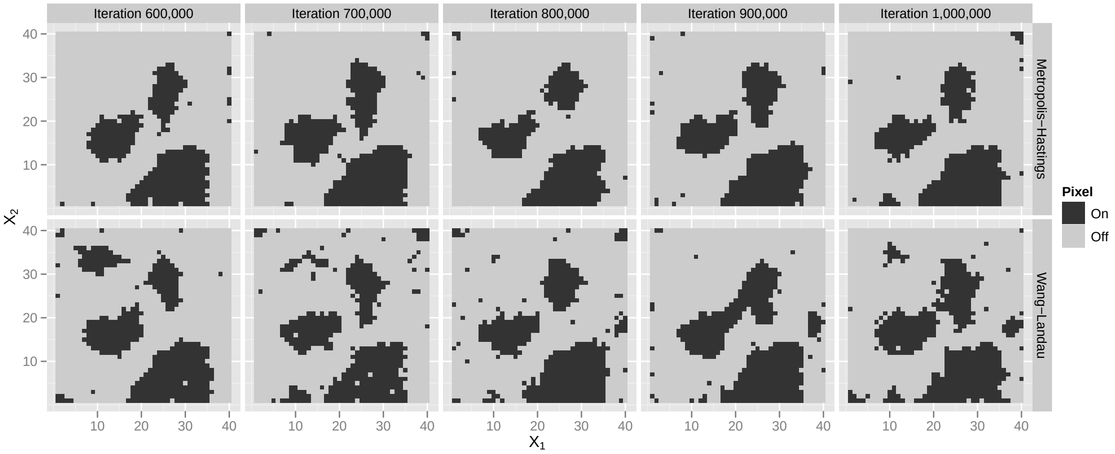


Figure 5.7: Spatial model example: states explored over 400,000 iterations for Metropolis-Hastings (top) and proposed algorithm (bottom).

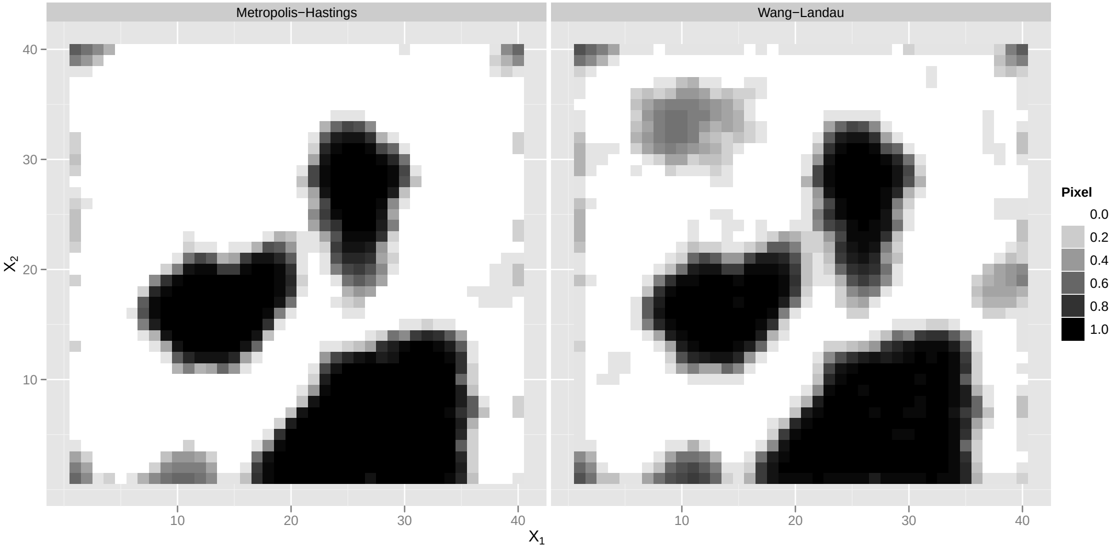


Figure 5.8: Spatial model example: average state explored with Metropolis-Hastings (left) and PAWL after importance sampling (right).

chain of each algorithm. We see that the proposed Wang-Landau algorithm encourages much more exploration of alternate modes. The corresponding average state explored over all 10 chains (after $400,000$ burn-in) is shown in Figure 5.8. From this we see that Wang-Landau induces exploration of the mode in the top-left of the region in question, as well as a bridge between the central ice floes. In conclusion, while flip-one-pixel Metropolis-Hastings is incapable of exploring the modes in the posterior caused by the presence/absence of large ice floes, the proposed algorithm encourages exploration of these modes, even in the presence of high between-pixel correlation. While [Higdon 98] develops a custom-tailored MCMC solution to overcome the inability of Metropolis-Hastings to adequately explore the posterior density in Ising models, we employ PAWL – a general-purpose automatic density exploration algorithm – to achieve similar results.

5.5 Discussion and Conclusion

The proposed algorithm, PAWL, has at its core the Wang-Landau algorithm which, despite wide-spread use in the physics community, has only recently been introduced into the statistics literature. A well-known obstacle in implementing the Wang-Landau algorithm is selecting the bins through which to discretize the state space; in response, we have developed a novel adaptive binning strategy. Additionally, we employ an adaptive proposal mechanism to further reduce the amount of user-defined parameters. Finally, to improve the convergence speed of the algorithm and to exploit modern computational power, we have developed a parallel interacting chain version of the algorithm which proves efficient in stabilizing the algorithm. Through a host of examples, we have demonstrated the algorithm's ability to conduct density exploration over a wide range of distributions continuous and discrete. While a suite of custom-purposed MCMC tools exist in the literature for each of these models, the proposed algorithm handles each within the same unified framework.

Further work will investigate the necessity of decreasing the step size γ_t. Indeed, if the interest lies in exploring the target distribution (and not in estimating the integrals $(\psi(i))_{i=1}^d$), then using a constant schedule $\gamma_t = 1$ throughout the algorithm already guarantees a desired frequency of visits in each bin, as seen in Chapter 4. Hence it is not clear whether or not decreasing the step size really stabilizes the exploration or, on the contrary, jeopardizes it. Moreover increasing the number of chains already stabilizes the estimates $(\theta_t(i))_{i=1}^d$; see Chapter 7 for further comments.

As practitioners in fields ranging from image-processing to astronomy turn to increasingly complex models to represent intricate real-world phenomena, the computational tools to approximate these models must grow accordingly. In this paper, we have proposed a general-purpose algorithm for automatic density exploration. Due to its fully adaptive nature, we foresee its application as a black-box exploratory MCMC method aimed at practitioners of Bayesian methods. While statisticians are well-accustomed to performing exploratory analysis in the modeling stage of an analysis, the notion of conducting preliminary general-purpose exploratory analysis in the Monte Carlo stage (or more generally, the model-fitting stage) of an analysis is an area which we feels deserves much further attention. As models grow in complexity, and endless model-specific Monte Carlo methods are proposed, it is valuable for the practitioner to have a universally applicable tool to throw at their problem before embarking on custom-tuned, hand-built Monte Carlo methods. Towards this aim, the authors have published an `R` package ("PAWL") to minimize user effort in applying the proposed algorithm to their specific problem.

Acknowledgements

Luke Bornn is supported by grants from NSERC and The Michael Smith Foundation for Health Research. Pierre E. Jacob is supported by a PhD fellowship from the AXA Research Fund. The authors also acknowledge the use of `R` and `ggplot2` in their analyses [R Development Core Team 10, Wickham 09]. Finally, the authors are thankful for helpful conversations with Yves Atchadé, François Caron, Nicolas Chopin, Faming Liang, Eric Moulines, and Christian Robert.

Bibliography

[Andrieu 05] C. Andrieu, E. Moulines & P. Priouret. *Stability of stochastic approximation under verifiable conditions.* SIAM Journal on control and optimization, vol. 44, no. 1, pages 283–312, 2005.

[Andrieu 06] C. Andrieu & E. Moulines. *On the ergodicity properties of some adaptive MCMC algorithms.* Annals of Applied Probability, vol. 16, no. 3, page 1462, 2006.

[Andrieu 08] C. Andrieu & J. Thoms. *A tutorial on adaptive MCMC.* Statistics and Computing, vol. 18, no. 4, pages 343–373, 2008.

[Atchadé 10] Y.F. Atchadé & JS Liu. *The Wang-Landau algorithm in general state spaces: applications and convergence analysis.* Statistica Sinica, vol. 20, pages 209–233, 2010.

[Atchadé 11] Y. Atchadé, G. Fort, E. Moulines & P. Priouret. Adaptive Markov chain Monte Carlo: Theory and methods, chapitre 2, pages 33–53. Cambridge University Press, Cambridge, UK, 2011.

[Banfield 92] J.D. Banfield & A.E. Raftery. *Ice floe identification in satellite images using mathematical morphology and clustering about principal curves.* Journal of the American Statistical Association, vol. 87, no. 417, pages 7–16, 1992.

[Baragatti 12] Meïli Baragatti, Agnès Grimaud & Denys Pommeret. *Parallel tempering with equi-energy moves.* Statistics and Computing, pages 1–17, 2012.

[Besag 93] J. Besag & P.J. Green. *Spatial statistics and Bayesian computation.* Journal of the Royal Statistical Society. Series B (Methodological), vol. 55, no. 1, pages 25–37, 1993.

[Brockwell 06] AE Brockwell. *Parallel Markov chain Monte Carlo simulation by pre-fetching.* Journal of Computational and Graphical Statistics, vol. 15, no. 1, pages 246–261, 2006.

[Brockwell 10] A. Brockwell, P. Del Moral & A. Doucet. *Sequentially interacting Markov chain Monte Carlo methods.* The Annals of Statistics, vol. 38, no. 6, pages 3387–3411, 2010.

[Byrd 10] J.M.R. Byrd. *Parallel Markov Chain Monte Carlo.* PhD thesis, University of Warwick, 2010.

[Casarin 11] R. Casarin, R. Craiu & F. Leisen. *Interacting multiple try algorithms with different proposal distributions.* Statistics and Computing, pages 1–16, 2011.

[Chopin 12] N. Chopin, T. Lelievre & G. Stoltz. *Free Energy Methods for Bayesian Statistics: Efficient Exploration of Univariate Gaussian Mixture Posteriors.* Statistics and Computing, vol. 22, no. 4, pages 897–916, 2012.

[Craiu 09] Radu V. Craiu, Jeffrey Rosenthal & Chao Yang. *Learn From Thy Neighbor: Parallel-Chain and Regional Adaptive MCMC.* Journal of the American Statistical Association, vol. 104, no. 488, pages 1454–1466, 2009.

[Del Moral 04] P Del Moral. Feynman-kac formulae. Springer, 2004.

[Del Moral 06] P. Del Moral, A. Doucet & A. Jasra. *Sequential Monte Carlo samplers.* Journal of the Royal Statistical Society: Series B, vol. 68, no. 3, pages 411–436, 2006.

[Diebolt 94] Jean Diebolt & Christian P. Robert. *Estimation of Finite Mixture Distributions through Bayesian Sampling.* Journal of the Royal Statistical Society. Series B (Methodological), vol. 56, no. 2, pages pp. 363–375, 1994.

[Edwards 88] R.G. Edwards & A.D. Sokal. *Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm.* Physical review D, vol. 38, no. 6, pages 2009–2012, 1988.

[Frühwirth-Schnatter 06] S. Frühwirth-Schnatter. Finite mixture and markov switching models. Springer, 2006.

[Geyer 91] C.J. Geyer. *Markov chain Monte Carlo Maximum Likelihood.* In E.M. Keramigas, editeur, Proceedings of the 23rd Symposium on the Interface, pages 156–163, Fairfax, 1991. Interface Foundations.

[Gilks 98] W.R. Gilks, G.O. Roberts & S.K. Sahu. *Adaptive Markov chain Monte Carlo through regeneration.* Journal of the American Statistical Association, vol. 93, no. 443, pages 1045–1054, 1998.

[Guan 07] Y. Guan & S.M. Krone. *Small-world MCMC and convergence to multi-modal distributions: From slow mixing to fast mixing.* The Annals of Applied Probability, vol. 17, no. 1, pages 284–304, 2007.

[Haario 01] H. Haario, E. Saksman & J. Tamminen. *An adaptive Metropolis algorithm.* Bernoulli, vol. 7, no. 2, pages 223–242, 2001.

[Higdon 98] D.M. Higdon. *Auxiliary Variable Methods for Markov Chain Monte Carlo with Applications.* Journal of the American Statistical Association, vol. 93, no. 442, 1998.

[Jacob 11] P. E. Jacob & R. J. Ryder. *The Wang-Landau algorithm reaches the Flat Histogram criterion in finite time.* ArXiv e-prints, October 2011.

[Jasra 05] A. Jasra, C. C. Holmes & D. A. Stephens. *MCMC and the label switching problem in Bayesian mixture models.* Statistical Science, vol. 20, pages 50–67, 2005.

[Jasra 07] A. Jasra, D.A. Stephens & C.C. Holmes. *On population-based simulation for static inference.* Statistics and Computing, vol. 17, no. 3, pages 263–279, 2007.

[Kou 06] SC Kou, Q. Zhou & W.H. Wong. *Equi-energy sampler with applications in statistical inference and statistical mechanics.* The Annals of Statistics, vol. 34, no. 4, pages 1581–1619, 2006.

[Lee 10] A. Lee, C. Yau, M.B. Giles, A. Doucet & C.C. Holmes. *On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods.* Journal of Computational and Graphical Statistics, vol. 19, no. 4, pages 769–789, 2010.

[Liang 05] F. Liang. *A Generalized Wang-Landau Algorithm for Monte Carlo Computation.* Journal of the American Statistical Association, vol. 100, no. 472, pages 1311–1327, 2005.

[Liang 07] F. Liang, C. Liu & R.J. Carroll. *Stochastic Approximation in Monte Carlo Computation.* Journal of the American Statistical Association, vol. 102, no. 477, page 305, 2007.

[Liang 09] F. Liang. *Improving SAMC using smoothing methods: Theory and applications to Bayesian model selection problems.* The Annals of Statistics, vol. 37, no. 5B, pages 2626–2654, 2009.

[Liang 10] F. Liang, C. Liu & R. Carrol. Advanced markov chain monte carlo methods. Wiley Online Library, 2010.

[Liang 11] F. Liang & M. Wu. *Population Stochastic Approximation MCMC Algorithm and its Weak Convergence.* Technical Report, Texas A& M University, 2011.

[Marin 07] J.M. Marin & C.P. Robert. Bayesian core: a practical approach to computational Bayesian statistics. Springer Verlag, 2007.

[Marinari 92] E. Marinari & G. Parisi. *Simulated tempering: a new Monte Carlo scheme.* EPL (Europhysics Letters), vol. 19, page 451, 1992.

[McDonald 73] G.C. McDonald & R.C. Schwing. *Instabilities of Regression Estimates Relating Air Pollution to Mortality.* Technometrics, vol. 15, no. 3, pages 463–481, 1973.

[Neal 01] R.M. Neal. *Annealed importance sampling.* Statistics and Computing, vol. 11, no. 2, pages 125–139, 2001.

[Neal 03] R.M. Neal. *Slice sampling.* Annals of Statistics, vol. 31, no. 3, pages 705–741, 2003.

[Niederreiter 92] H. Niederreiter. Random number generation and quasi-Monte Carlo methods. Society for Industrial Mathematics, 1992.

[R Development Core Team 10] R Development Core Team. *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria, 2010.

[Richardson 97] Sylvia. Richardson & Peter J. Green. *On Bayesian Analysis of Mixtures with an Unknown Number of Components (with discussion)*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 59, no. 4, pages 731–792, 1997.

[Robert 04] C.P. Robert & G. Casella. Monte carlo statistical methods. Springer, 2004.

[Roberts 97] G.O. Roberts, A. Gelman & W.R. Gilks. *Weak convergence and optimal scaling of random walk Metropolis algorithms*. The Annals of Applied Probability, vol. 7, no. 1, pages 110–120, 1997.

[Roberts 09] G.O. Roberts & J.S. Rosenthal. *Examples of Adaptive MCMC*. Journal of Computational and Graphical Statistics, vol. 18, pages 349–367, 2009.

[Schmidler 11] Scott Schmidler. *Exploration vs. Exploitation in Adaptive MCMC*. Adap'ski Invited Presentation, 2011.

[Stephens 00] M. Stephens. *Dealing with label switching in mixture models*. Journal of the Royal Statistical Society: Series B, vol. 62(4), pages 795–809, 2000.

[Suchard 09] M.A. Suchard & A. Rambaut. *Many-core algorithms for statistical phylogenetics*. Bioinformatics, vol. 25, no. 11, page 1370, 2009.

[Swendsen 86] R.H. Swendsen & J.S. Wang. *Replica Monte Carlo simulation of spin-glasses*. Physical Review Letters, vol. 57, no. 21, pages 2607–2609, 1986.

[Swendsen 87] R.H. Swendsen & J.S. Wang. *Nonuniversal critical dynamics in Monte Carlo simulations*. Physical Review Letters, vol. 58, no. 2, pages 86–88, 1987.

[Wang 01a] F. Wang & DP Landau. *Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram*. Physical Review E, vol. 64, no. 5, page 56101, 2001.

[Wang 01b] F. Wang & DP Landau. *Efficient, multiple-range random walk algorithm to calculate the density of states*. Physical Review Letters, vol. 86, no. 10, pages 2050–2053, 2001.

[Wei 04] W. Wei, J. Erenrich & B. Selman. *Towards efficient sampling: Exploiting random walk strategies*. In Proceedings of the National Conference on Artificial Intelligence, pages 670–676. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.

[Welford 62] BP Welford. *Note on a method for calculating corrected sums of squares and products*. Technometrics, vol. 4, no. 3, pages 419–420, 1962.

[Wickham 09] Hadley Wickham. ggplot2: Elegant graphics for data analysis. Springer New York, 2009.

[Zellner 86] A. Zellner. *On assessing prior distributions and Bayesian regression analysis with g-prior distributions.* In P.K. Goel & A. Zellner, editeurs, Bayesian Inference and Decision Techniques: Essays in Honor of Bruno do Dinetti, pages 233–243. North-Holland, 1986.

A Details of Proposed Algorithm

In Algorithm 9 we detail PAWL, fusing together a Wang-Landau base with adaptive binning, interacting parallel chains, and an adaptive proposal mechanism. In comparison to the generalized Wang-Landau algorithm (Algorithm 8), when a flat histogram is reached the distribution of particles within bins is tested to determine whether a given bin should be split. In addition, a suite of N interacting chains is employed, and hence the former chain $\boldsymbol{X}_t$ is now made of N chains: $\boldsymbol{X}_t = (X_t^{(1)}, \dots, X_t^{(N)})$, each defined on the state space $\mathcal{X}$. All the N chains are used to update the bias θ_t, as described in Section 5.3.3.

The chains are moved using an adaptive mechanism determined by the Metropolis-Hastings acceptance rate as explained in Section 5.3.4. While we present Algorithm 9 with adaptive proposal variance, it may also be implemented with an adaptive mixture proposal as described in Section 5.3.4. Note that when a bin is split, it is possible to set the desired frequency of the new bins to some reduced value, say each obtaining half the desired frequency of the original – in fact in the numerical experiments we do exactly that. However, for notational and pedagogical simplicity, we present here the algorithm where the desired frequency of each bin is equal to $1/d_t$ at iteration t.

B Algorithm Convergence

The convergence of the Wang-Landau algorithm using a deterministic stepsize, also called the Stochastic Approximation Monte Carlo (SAMC) algorithm, and stochastic approximation algorithms in general, has been well-studied in [Atchadé 10, Andrieu 06, Andrieu 05, Liang 11], see Chapter 7 of [Liang 10] for a recent introduction. Since writing this manuscript, we have also learned of recent convergence and ergodicity results for a parallel implementation of the SAMC algorithm [Liang 11]. However, as noted in [Liang 11] these results fail to explain why the parallel version is more efficient than the single-chain algorithm in practice; instead it proves the consistency of the algorithm when the number of iterations goes to infinity, and the asymptotic normality of the bias $(\theta_t)_{t\geq 0}$, for any fixed number of chains. We believe that precise statements on the impact of the number of chains upon the stabilization of the bias $(\theta_t)_{t\geq 0}$ would require the analysis of the Feynman–Kac semigroup associated with the algorithm, similar to what is commonly used to study the impact of the number of particles in Sequential Monte Carlo methods [Del Moral 04].

Each of our proposed improvements adds a level of complexity to the proof of the algorithm's consistency. First and foremost, we are using the Flat Histogram criterion, and thus the usual assumptions on the stepsize of the stochastic approximation are not easily verified (e.g. assumptions of Theorem 2.3 in [Andrieu 05] and conditions (A4) in [Liang 11]). Indeed, if no flat histogram criterion was met, then the stepsize $(\gamma_k)_{k\geq 0}$ would stay constant. We rely on a result in [Jacob 11] that proves that the criterion is met in finite time, for any precision threshold c; therefore the results of [Andrieu 05] and thus the results of [Liang 11] apply even when one uses the flat histogram criterion.

Finally, with our inclusion of an adaptive proposal as a Robbins-Monro style update, the algorithm still remains in the class of stochastic approximation algorithms. One could pragmatically stop the adaptation of the proposal distribution after some iteration and fall back to the study of a homogeneous Metropolis–Hastings algorithm. However, we believe that the algorithm could be studied in the same framework as [Andrieu 06, Liang 11], where now the stochastic approximation would both control the bias $(\theta_t)_{t\geq 0}$ and the standard deviation of the proposal $(\sigma_t)_{t\geq 0}$.

Algorithm 9 Proposed Density Exploration Algorithm

```
Run a preliminary exploration of the target e.g. using adaptive MCMC, and determine an energy range.
Partition the state space into $d_0$ regions $\{\mathcal{X}_{1,0}, \dots, \mathcal{X}_{d_0,0}\}$ along a reaction coordinate $\xi(x)$,
the default choice being $\xi(x) = -\log \pi(x)$.
$\forall i \in \{1, \dots, d_0\}$ set $\theta(i) \leftarrow 1, \nu(i) \leftarrow 0$.
Choose an initial proposal standard deviation $\sigma_0$
Choose the frequency $\tau$ with which to check for a flat histogram.
Choose a decreasing sequence $\{\rho_t\}$, typically $\rho_t = 1/t$, to update the proposal standard deviation.
Choose a decreasing sequence $\{\gamma_k\}$, typically $\gamma_k = 1/k$, to update the bias.
Sample $\boldsymbol{X}_0 \sim \pi_0$, an initial distribution.
for $t = 1$ to $T$ do
    Sample $\boldsymbol{X}_t$ from $P_{\theta_{t-1}}(\boldsymbol{X}_{t-1}, \cdot)$, a transition kernel with invariant distribution $\prod_{n=1}^{N} \tilde{\pi}_{\theta_{t-1}}(x)$, parametrized by the proposal standard deviation $\sigma_{t-1}$.
    Update the proposal standard deviation:
        $$\sigma_t \leftarrow \sigma_{t-1} + \rho_t \left(2\mathcal{I}(A > .234) - 1\right)$$
    , where $A$ is the last acceptance rate.
    Set $d_t \leftarrow d_{t-1}$.
    Update the proportions:
        $$\forall i \in \{1, \dots, d_t\} \quad \nu(i) \leftarrow \frac{1}{t}\left[(t-1)\nu(i) + N^{-1}\sum_{j=1}^{N} \mathcal{I}_{\mathcal{X}_{i,t}}(X_t^{(j)})\right]$$
    Every $\tau$-th iteration, check the distribution of samples within each bin, extending the range if necessary. For example, if if $\xi(x) = -\log \pi(x)$ and a new minimum value of $\xi(x)$ was found, extend the first bin in order to include this value.
    for $i \in \{1, \dots, d_t\}$ do
        if bin $i$ should be split then
            Create two sub-bins covering bin $i$, assign to each a weight equal to $\theta_t(i)/2$.
            Set $d_t \leftarrow d_t + 1$, extend $\nu$.
        end if
    end for
    if "flat histogram": $\max_{i \in \{1,\dots,d_t\}} |\nu(i) - {d_t}^{-1}| < c/d$ then
        Set $k \leftarrow k + 1$.
        Reset $\forall i \in \{1, \dots, d_t\} \quad \nu(i) \leftarrow 0$.
    end if
    Update the bias: $\log \theta_t(i) \leftarrow \log \theta_{t-1}(i) + \gamma_k(N^{-1} \sum_{j=1}^{N} \mathcal{I}_{\mathcal{X}_{i,t}}(X_t^{(j)}) - d_t^{-1})$.
    Normalize the bias: $\theta_t(i) \leftarrow \theta_t(i) / \sum_{i=1}^{d_t} \theta_t(i)$.
end for
```

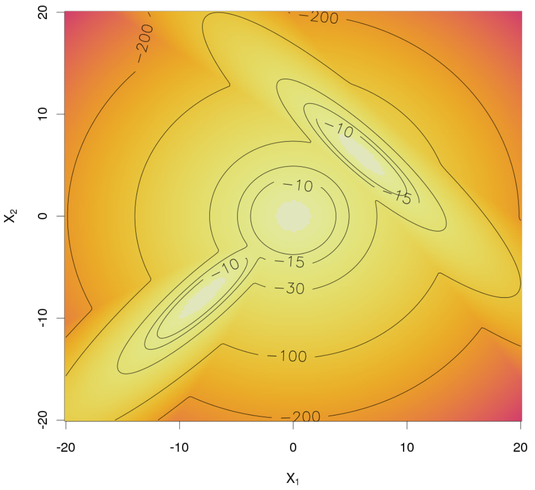


Figure 9: Trimodal example: log density function of the target distribution (4). The modes are separated by areas where the log density is very low, making exploration difficult.

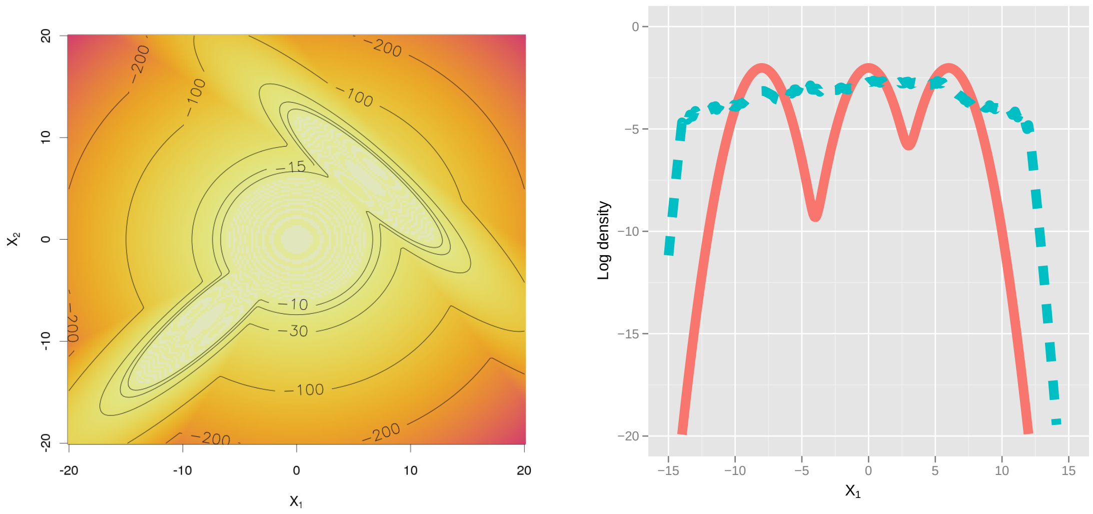
C Trimodal Target Example

We introduce a toy target distribution to aid in demonstrating some of the concepts discussed earlier, especially the bin splitting strategy. Consider the 2-dimensional trimodal target described in [Liang 07]:

$$X \sim \frac{1}{3}N\left[\begin{pmatrix} 8 \\ 8 \end{pmatrix}, \begin{pmatrix} 1 & .9 \\ .9 & 1 \end{pmatrix}\right] + \frac{1}{3}N\left[\begin{pmatrix} 6 \\ 6 \end{pmatrix}, \begin{pmatrix} 1 & -.9 \\ -.9 & 1 \end{pmatrix}\right] + \frac{1}{3}N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right], \tag{4}$$

a mixture of three bivariate Gaussian distributions. The corresponding log density is shown in Figure 9. This density, while low-dimensional and with only three modes, is known to be difficult to sample from with Metropolis-Hastings (e.g. [Gilks 98]). Firstly, with different correlation structures in each mode, an adaptive algorithm might conform to one mode, missing (or poorly sampling from) the other two. Secondly, there is a low-density region separating each mode; as such, low-variance proposals might be incapable of jumping between modes. Figure 10 displays the biased target (Equation (1), using the log density as reaction coordinate) and one of its marginals, emphasizing the effect of biasing in improving the ability for the algorithm to explore the density. Here the plot is created using computationally expensive fine-scale numerical integration.

Initial exploration is performed by an adaptive MCMC algorithm, run with 2 parallel chains and 500 iterations. The proposal of this first run targets a specific acceptance rate of 23.4%, as described in Section 3.4. The explored energy range is expanded and divided into $d_0 = 3$ initial bins. In all examples, we use $c = 0.5$. The proposed algorithm is run with 2 parallel chains for 2500 iterations. Figure 11(a) shows the regions recovered by the chains; the chains have moved easily between the modes, even if the distribution of the starting point was not well spread over the target density support. In this case, to reflect the possible lack of information about the target support, we draw the starting points of all the chains from a $\mathcal{N}(0, 0.1 \times I_2)$ distribution, hence exclusively in one of the three modes. In this setting the free energy SMC method described in Chapter 3 fails to recover the target distribution accurately; specifically, the central mode is over-sampled due to many particles not reaching the outer modes. However, if the initial distribution π_0 is well spread over the target support, the SMC algorithm recovers the modes. Figure 11(b) shows the points generated by 3000 iterations of the adaptive Metropolis–Hastings algorithm (already used in the initial exploration), also using 2 chains. We see that the



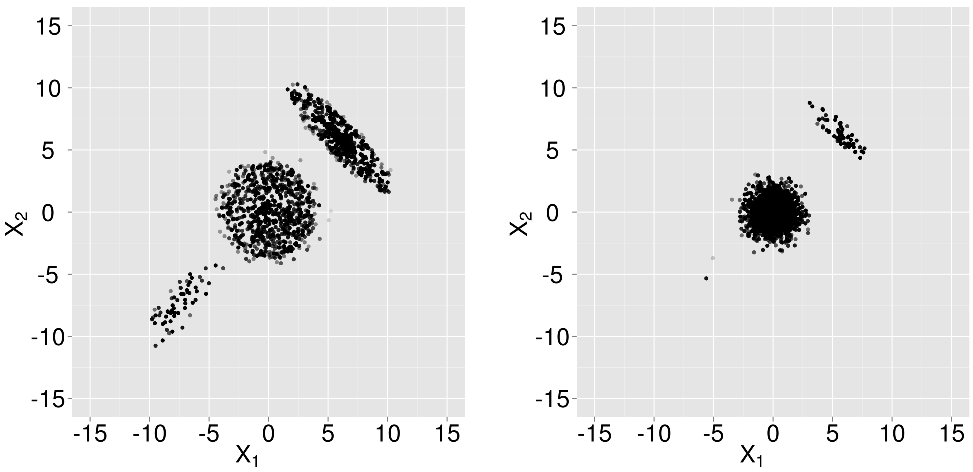
(a) Biased target density

(b) Marginal of biased and unbiased target density

Figure 10: Trimodal example: (left) log density of the biased target distribution (Equation (1)). The former modes are now all located in a fairly flat region, allowing for straightforward exploration. (Right) marginal log density of one component of the (symmetric) trimodal target. The solid line shows the target probability density function and the dashed line shows an approximation of the marginal of the biased distribution of Equation (1). The biased marginal is flatter, hence easier to explore than the original target distribution.

exploration was less successful, with the bottom left mode hardly visited at all, although the same number of point-wise density evaluations were performed as for the proposed algorithm.

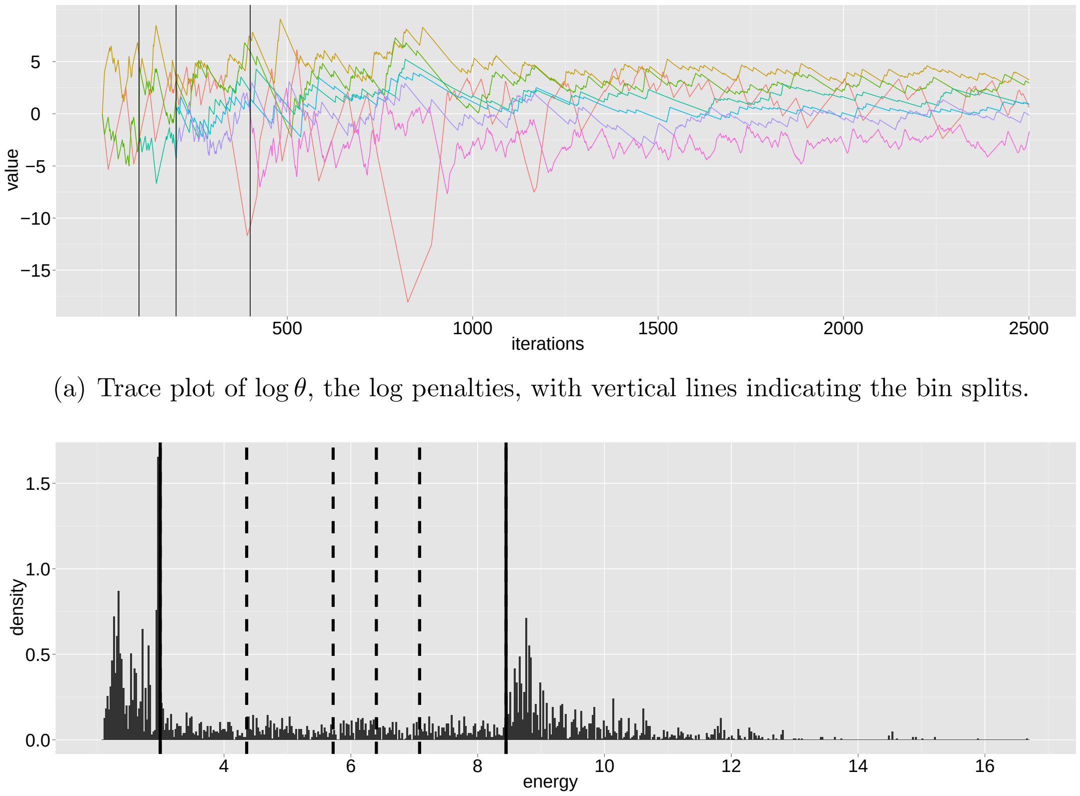
Along the iterations, the bins have been split three times. Here the chosen strategy was to split a bin if less than 25% of its points were situated in half of the bin. Figure 12 illustrates the effects of the binning strategy. Figure 12(a) shows the trace plot of the estimators θ, and the iterations at which bins are split are shown with vertical lines. After each split, the dimension of θ increases but the figure shows that the new estimators quickly stabilize. After the last split around iteration 450, the number of bins stays constant. Figure 12(b) shows the histogram of the log density evaluations of the chain points, with vertical full lines showing the initial bins, and vertical dashed lines showing the bins that have been added during the run. We see that the bin splits induce more uniformity within bins, and hence across the entire reaction coordinate range, aiding in movement and exploration.



(a) Scatter plot of the chains generated by the proposed algorithm

(b) Scatter plot of the chains generated by an adaptive Metropolis–Hastings algorithm

Figure 11: Trimodal example: results of the proposed algorithm: (a) Scatter plot of all samples, before normalization/importance sampling, (b) Scatter plot of samples generated by an adaptive Metropolis–Hastings algorithm using the same number of chains and iterations.



(a) Trace plot of $\log\theta$, the log penalties, with vertical lines indicating the bin splits.

(b) Histogram of the energy values computed during the algorithm. The vertical full lines show the initial bins and dashed lines show the cuts that have been added dynamically.

Figure 12: Trimodal example: histograms of the log density values of all the chain points just before the iterations at which the splitting mechanism is triggered. The number of bins increases automatically along the iterations.

Chapter 6

SMC^2 for sequential analysis of state-space models

Ce chapitre présente une méthode de Monte Carlo séquentiel pour simuler selon la loi *a posteriori* des paramètres et des états latents dans les modèles à chaîne de Markov cachée (aussi appelés modèles à espace d'états).

La méthode est une combinaison de la méthode de Monte Carlo séquentiel pour l'échantillonnage dans les problèmes statiques, et de la méthode de Monte Carlo séquentiel pour l'échantillonnage des lois de filtrage dans les modèles à chaîne de Markov cachée. Appelée SMC^2, elle permet ainsi l'analyse bayésienne de ces modèles de manière séquentielle: lorsqu'un nouveau lot d'observations est disponible, l'inférence peut être mise à jour sans nécessiter de relancer intégralement l'algorithme. Ainsi, elle permet d'observer l'évolution des lois *a posteriori* au fur et à mesure que les observations arrivent. Les étapes de réjuvénation, nécessaires à tout algorithme de Monte Carlo séquentiel appliqué aux problèmes statiques, sont assurées par des noyaux de Markov introduits récemment par les méthodes appelées Monte Carlo à chaîne de Markov particulaires.

Par ailleurs le nombre de particules utilisées peut être augmenté dynamiquement au cours de l'algorithme, ce qui aboutit à une méthode facile à régler pour l'utilisateur. Enfin l'algorithme permet d'estimer l'évidence, une quantité clef pour le choix de modèle bayésien. La méthode n'est toutefois pas *en ligne*, puisque son coût computationnel augmente avec le nombre d'observations, en raison des étapes de réjuvénation. Ce coût est étudié en détail, et sous certaines hypothèses il peut être borné par une quantité proportionnelle au carré du nombre d'observations.

À travers deux exemples pratiques, les performances de l'algorithme sont illustrées et comparées à celles d'autres algorithmes existants. Le premier exemple est un modèle de volatilité stochastique où l'état latent est un processus de Lévy discrétisé, appliqué aux données S&P 500. Le second est un modèle où les observations suivent une séquence de lois généralisées de valeurs extrêmes, et est appliqué à un jeu de données de records athlétiques.

Ce chapitre s'accompagne d'une bibliothèque de fonctions écrite en langage `python`, appelée `py-smc2`, qui permet de comparer la méthode proposée à différentes méthodes concurrentes. La bibliothèque est disponible à l'adresse suivante:

http://code.google.com/p/py-smc2/

Authors
- Nicolas Chopin (CREST–ENSAE, Paris)
- Pierre E. Jacob (Université Paris-Dauphine, CREST, Paris)
- Omiros Papaspiliopoulos (Universitat Pompeu Fabra, Barcelona)

Status To be published in the *Journal of the Royal Statistical Society: Series B*, arXiv number 1101.1528.

Abstract

We consider the generic problem of performing sequential Bayesian inference in a state-space model with observation process y, state process x and fixed parameter θ. An idealized approach would be to apply the iterated batch importance sampling (IBIS) algorithm of Chopin (2002). This is a sequential Monte Carlo algorithm in the θ-dimension, that samples values of θ, reweights iteratively these values using the likelihood increments $p(y_t|y_{1:t-1},\theta)$, and rejuvenates the θ-particles through a resampling step and a MCMC update step. In state-space models these likelihood increments are intractable in most cases, but they may be unbiasedly estimated by a particle filter in the x-dimension, for any fixed θ. This motivates the SMC^2 algorithm proposed in this article: a sequential Monte Carlo algorithm, defined in the θ-dimension, which propagates and resamples many particle filters in the x-dimension. The filters in the x-dimension are an example of the random weight particle filter as in Fearnhead et al. (2010). On the other hand, the particle Markov chain Monte Carlo (PMCMC) framework developed in Andrieu et al. (2010) allows us to design appropriate MCMC rejuvenation steps. Thus, the θ-particles target the correct posterior distribution at each iteration t, despite the intractability of the likelihood increments. We explore the applicability of our algorithm in both sequential and non-sequential applications and consider various degrees of freedom, as for example increasing dynamically the number of x-particles. We contrast our approach to various competing methods, both conceptually and empirically through a detailed simulation study, included here and in a supplement, and based on particularly challenging examples.

Keywords: Iterated batch importance sampling; Particle filtering; Particle Markov chain Monte Carlo; Sequential Monte Carlo; State-space models

6.1 Introduction

6.1.1 Objectives

We consider a generic state-space model, with parameters $\theta \in \Theta$, prior $p(\theta)$, latent Markov process (x_t), $p(x_1|\theta) = \mu_\theta(x_1)$,

$$p(x_{t+1}|x_{1:t},\theta) = p(x_{t+1}|x_t,\theta) = f_\theta(x_{t+1}|x_t), \quad t \geq 1,$$

and observed process

$$p(y_t|y_{1:t-1},x_{1:t-1},\theta) = p(y_t|x_t,\theta) = g_\theta(y_t|x_t), \quad t \geq 1.$$

For an overview of such models with references to a wide range of applications in Engineering, Economics, Natural Sciences, and other fields, see e.g. [Doucet 01], [Künsch 01] or [Cappé 05].

We are interested in the recursive exploration of the sequence of posterior distributions

$$\pi_0(\theta) = p(\theta), \quad \pi_t(\theta, x_{1:t}) = p(\theta, x_{1:t}|y_{1:t}), \quad t \geq 1\,, \tag{6.1}$$

as well as computing the model evidence $p(y_{1:t})$ for model composition. Such a sequential analysis of state-space models under parameter uncertainty is of interest in many settings; a simple example is out-of-sample prediction, and related goodness-of-fit diagnostics based on prediction residuals, which are popular for instance in Econometrics; see e.g. Section 4.3 of [Kim 98] or [Koop 07]. Furthermore, we shall see that recursive exploration up to time $t = T$ may be computationally advantageous even in batch estimation scenarios, where a fixed observation record $y_{1:T}$ is available.

6.1.2 State of the art

Sequential Monte Carlo (SMC) methods are considered the state of the art for tackling this kind of problems. Their appeal lies in the efficient re-use of samples across different times t, compared for example with MCMC methods which would typically have to be re-run for each time horizon. Additionally, convergence properties (with respect to the number of simulations) under mild assumptions are now well understood; see e.g. [Del Moral 99], [Crisan 02], [Chopin 04], [Oudjane 05], [Douc 08]. See also [Del Moral 06] for a recent overview of SMC methods.

SMC methods are particularly (and rather unarguably) effective for exploring the simpler sequence of posteriors, $\pi_t(x_t|\theta) = p(x_t|y_{1:t}, \theta)$; compared to the general case the static parameters are treated as known and interest is focused on x_t as opposed to the whole path $x_{0:t}$. This is typically called the filtering problem. The corresponding algorithms are known as *particle filters* (PFs); they are described in Section 6.2.1 in some detail. These algorithms evolve, weight and resample a population of N_x number of particles, $x_t^{1:N_x}$, so that at each time t they are a properly weighted sample from $\pi_t(x_t|\theta)$. Recall that a particle system is called properly weighted if the weights associated with each sample are unbiased estimates of the Radon-Nikodym derivative between the target and the proposal distribution; see for example Section 1 of [Fearnhead 10a] and references therein. A by-product of the PF output is an unbiased estimator of the likelihood increments and the marginal likelihood

$$p(y_{1:t}|\theta) = p(y_1|\theta) \prod_{s=2}^{t} p(y_s|y_{1:s-1}, \theta), \quad 1 \leq t \leq T. \tag{6.2}$$

the variance of which increases linearly over time [Cérou 11].

Complementary to this setting is the iterated batch importance sampling (IBIS) algorithm of [Chopin 02] for the recursive exploration of the sequence of parameter posterior distributions, $\pi_t(\theta)$; the algorithm is outlined in Section 6.2.2. This is also an SMC algorithm which updates a population of N_θ particles, $\theta^{1:N_\theta}$, so that at each time t they are a properly weighted sample from $\pi_t(\theta)$. The algorithm includes occasional MCMC steps for rejuvenating the current population of θ-particles to prevent the number of distinct θ-particles from decreasing over time. Implementation of the algorithm requires the likelihood increments $p(y_t|y_{1:t-1}, \theta)$ to be computable. This constrains the application of IBIS in state-space models since computing the increments involves integrating out the latent states. Notable exceptions are linear Gaussian state-space models and models where x_t takes values in a finite set. In such cases a Kalman filter and a Baum filter respectively can be associated to each θ-particle to evaluate efficiently the likelihood increments; see e.g. [Chopin 07].

On the other hand, sequential inference for both parameters and latent states for a generic state-space model is a much harder problem, which, although very important in applications, is still rather unresolved; see for example [Doucet 11], [Andrieu 10], [Doucet 09] for recent discussions. The batch estimation problem of exploring $\pi_T(\theta, x_{0:T})$ is a non-trivial MCMC problem on its own right, especially for large T. This is due to both high dependence between parameters and the latent process, which affects Gibbs sampling strategies [Papaspiliopoulos 07], and the difficulty in designing efficient simulation schemes for sampling from $\pi_T(x_{0:T}|\theta)$. To address these problems [Andrieu 10] developed a general theory of particle Markov chain Monte Carlo (PMCMC) algorithms, which are MCMC algorithms that use a particle filter of size N_x as a proposal mechanism. Superficially, it appears that the algorithm replaces the intractable (6.2) by the unbiased estimator provided by the PF within an MCMC algorithm that samples from $\pi_T(\theta)$. However, [Andrieu 10] show that (a) as N_x grows, the PMCMC algorithm behaves more and more like the theoretical MCMC algorithm which targets the intractable $\pi_T(\theta)$; and (b) for any fixed value of N_x, the PMCMC algorithm admits $\pi_T(\theta, x_{0:T})$ as a stationary distribution. The exactness (in terms of not perturbing the stationary distribution) follows from demonstrating that the PMCMC is an ordinary MCMC algorithm (with specific proposal distributions) on an expanded model which includes the PF as auxiliary variables; when $N_x = 1$ this augmentation collapses to the more familiar scheme of imputing the latent states.

6.1.3 Proposed algorithm

SMC^2 is a generic black box tool for performing sequential analysis of state-space models, which can be seen as a natural extension of both IBIS and PMCMC. To each of the N_θ $\theta-$particles θ^m, we attach a PF which propagates N_x $x-$particles; due to the nested filters we call it the SMC^2 algorithm. Unlike the implementation of IBIS which carries an exact filter, in this case the PFs only produce unbiased estimates of the marginal likelihood. This ensures that the θ-particles are properly weighted for $\pi_t(\theta)$, in the spirit of the *random weight PF* of e.g. [Fearnhead 10a]. The connection with the auxiliary representation underlying PMCMC is pivotal for designing the MCMC rejuvenation steps, which are crucial for the success of IBIS. We obtain a sequential auxiliary Markov representation, and use it to formally demonstrate that our algorithm explores the sequence defined in (6.1). The case $N_x = \infty$ corresponds to an (unrealisable) IBIS algorithm, whereas $N_x = 1$ to an importance sampling scheme, the variance of which typically grows polynomially with t [Chopin 04].

SMC^2 is a sequential but not an on-line algorithm. The computational load increases with iterations due to the associated cost of the MCMC steps. Nevertheless, these steps typically occur at a decreasing rate (see Section 3.8 for details). The only on-line generic algorithm for sequential analysis of state-space models we are aware of is the self-organizing particle filter (SOPF) of [Kitagawa 98]: this is PF applied to the extended state $\tilde{x}_t = (x_t, \theta)$, which never updates the θ-component of particles, and typically diverges quickly over time [e.g. Doucet 09]; see also [Liu 01] for a modification of SOPF which we discuss later. Thus, a genuinely on-line analysis, which would provide constant Monte Carlo error at a constant CPU cost, with respect to all the components of $(x_{1:t}, \theta)$ may well be an unattainable goal. This is unfortunate, but hardly surprising, given that the target π_t is of increasing dimension. For certain models with a specific structure (e.g the existence of sufficient statistics), an on-line algorithm may be obtained by extending SOPF so as to include MCMC updates of the θ-component, see [Gilks 01], [Fearnhead 02], [Storvik 02], and also the more recent work of [Carvalho 10], but numerical evidence seems to indicate these algorithms degenerate as well, albeit possibly at a slower rate; see e.g. [Doucet 09]. On the other hand, SMC^2 is a generic approach which does not require such a specific structure.

Even in batch estimation scenarios SMC^2 may offer several advantages over PMCMC, in the same way that SMC approaches may be advantageous over MCMC methods [Neal 01, Chopin 02, Cappé 04, Del Moral 06, Jasra 07]. Under certain conditions (which relate to the asymptotic normality of the maximizer of (6.2)) SMC^2 has the same complexity as PMCMC. Nevertheless, it calibrates automatically its tuning parameters, as for example N_x and the proposal distributions for θ. (Note adaptive versions of PMCMC, see e.g. [Silva 09] and [Peters 10] exist however.) Then, the first iterations of the SMC^2 algorithm make it possible to quickly discard uninteresting parts of the sampling space, using only a small number of observations. Finally, the SMC^2 algorithm provides an estimate of the evidence (marginal likelihood) of the model $p(y_{1:T})$ as a direct by-product.

We demonstrate the potential of the SMC^2 on two classes of problems which involve multidimensional state processes and several parameters: volatility prediction for financial assets using Lévy driven stochastic volatility models, and likelihood assessment of athletic records using time-varying extreme value distributions. A supplement to this article (available on the third author's web-page) contains further numerical investigations with the SMC^2 and competing methods on more standard examples.

Finally, it has been pointed to us that [Fulop 11b] have developed independently and concurrently an algorithm similar to SMC^2. Distinctive features of our paper are the generality of the proposed approach, so that it may be used more or less automatically on complex examples (e.g. setting N_x dynamically), and the formal results that establish the validity of the SMC^2 algorithm, and its complexity.

6.1.4 Plan, notations

The paper is organised as follows. Section 6.2 recalls the two basic ingredients of SMC^2: the PF and the IBIS. Section 6.3 introduces the SMC^2 algorithm, provides its formal justification, discusses its complexity and the latitude in its implementation. Section 6.4 carries out a detailed simulation study which investigates the performance of SMC^2 on particularly challenging models. Section 6.5 concludes.

As above, we shall use extensively the concise colon notation for sets of random variables, e.g. $x_t^{1:N_x}$ is a set of N_x random variables x_t^n, $n = 1, \ldots, N_x$, $x_{1:t}^{1:N_x}$ is the union of the sets $x_s^{1:N_x}$, $s = 1, \ldots, t$, and so on. In the same vein, $1 : N_x$ stands for the set $\{1, \ldots, N_x\}$. Particle (resp. time) indices are always in superscript (resp. subscript). The letter p refers to probability densities defined by the model, e.g. $p(\theta)$, $p(y_{1:t}|\theta)$, while π_t refers to the probability density targeted at time t by the algorithm, or the corresponding marginal density with respect to its arguments.

6.2 Preliminaries

6.2.1 Particle filters (PFs)

We describe a particle filter that approximates recursively the sequence of filtering densities $\pi_t(x_t|\theta) = p(x_t|y_{1:t}, \theta)$, for a fixed parameter value θ. The formalism is chosen with view to integrating this algorithm into SMC^2. We first give a pseudo-code version, and then we detail the notations. Any operation involving the superscript n must be understood as performed for $n \in 1 : N_x$, where N_x is the total number of particles.

Step 1: At iteration $t = 1$,

(a) Sample $x_1^n \sim q_{1,\theta}(\cdot)$.

(b) Compute and normalise weights

$$w_{1,\theta}(x_1^n) = \frac{\mu_\theta(x_1^n) g_\theta(y_1|x_1^n)}{q_{1,\theta}(x_1^n)}, \quad W_{1,\theta}^n = \frac{w_{1,\theta}(x_1^n)}{\sum_{i=1}^{N_x} w_{1,\theta}(x_1^i)}.$$

Step 2: At iteration $t = 2 : T$,

(a) Sample the index $a_{t-1}^n \sim \mathcal{M}(W_{t-1,\theta}^{1:N_x})$ of the ancestor of particle n.

(b) Sample $x_t^n \sim q_{t,\theta}(\cdot|x_{t-1}^{a_{t-1}^n})$.

(c) Compute and normalise weights

$$w_{t,\theta}(x_{t-1}^{a_{t-1}^n}, x_t^n) = \frac{f_\theta(x_t^n|x_{t-1}^{a_{t-1}^n}) g_\theta(y_t|x_t^n)}{q_{t,\theta}(x_t^n|x_{t-1}^{a_{t-1}^n})}, \quad W_{t,\theta}^n = \frac{w_{t,\theta}(x_{t-1}^{a_{t-1}^n}, x_t^n)}{\sum_{i=1}^{N_x} w_{t,\theta}(x_{t-1}^{a_{t-1}^i}, x_t^i)}.$$

In this algorithm, $\mathcal{M}(W_{t-1,\theta}^{1:N_x})$ stands for the multinomial distribution which assigns probability $W_{t-1,\theta}^n$ to outcome $n \in 1 : N_x$, and $(q_{t,\theta})_{t \in 1:T}$ stands for a sequence of conditional proposal distributions which depend on θ. A standard, albeit sub-optimal, choice is the prior, $q_{1,\theta}(x_1) = \mu_\theta(x_1)$, $q_{t,\theta}(x_t|x_{t-1}) = f_\theta(x_t|x_{t-1})$ for $t \geq 2$, which leads to the simplification

$$w_{t,\theta}(x_{t-1}^{a_{t-1}^n}, x_t^n) = g_\theta(y_t|x_t^n).$$

We note in passing that Step (a) is equivalent to multinomial resampling [e.g. Gordon 93]. Other, more efficient schemes exist [Liu 98, Kitagawa 98, Carpenter 99], but are not discussed in the paper for the sake of simplicity.

At iteration t, the following quantity

$$\frac{1}{N_x}\sum_{n=1}^{N_x} w_{t,\theta}(x_{t-1}^{a_{t-1}^n}, x_t^n)$$

is an unbiased estimator of $p(y_t|y_{1:t-1},\theta)$. More generally, it is a key feature of PFs that

$$\hat{Z}_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}) = \left(\frac{1}{N_x}\right)^t \left\{\sum_{n=1}^{N_x} w_{1,\theta}(x_1^n)\right\} \prod_{s=2}^{t} \left\{\sum_{n=1}^{N_x} w_{s,\theta}(x_{s-1}^{a_{s-1}^n}, x_s^n)\right\} \tag{6.3}$$

is also an unbiased estimator of $p(y_{1:t}|\theta)$; this is not a straightforward result, see Proposition 7.4.1 in [Del Moral 04]. We denote by $\psi_{1,\theta}(x_1^{1:N_x})$, for $t = 1$, and $\psi_{t,\theta}(x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})$ for $t \geq 2$, the joint probability density of all the random variables generated during the course of the algorithm up to iteration t. Thus, the expectation of the random variable $\hat{Z}_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})$ with respect to $\psi_{t,\theta}$ is exactly $p(y_{1:t}|\theta)$.

6.2.2 Iterated batch importance sampling (IBIS)

The IBIS approach of [Chopin 02] is an SMC algorithm for exploring a sequence of parameter posterior distributions $\pi_t(\theta) = p(\theta|y_{1:t})$. All the operations involving the particle index m must be understood as operations performed for all $m \in 1 : N_\theta$, where N_θ is the total number of θ-particles.

Sample θ^m from $p(\theta)$ and set $\omega^m \leftarrow 1$. Then, at time $t = 1 : T$

(a) Compute the incremental weights and their weighted average

$$u_t(\theta^m) = p(y_t|y_{1:t-1},\theta^m), \quad L_t = \frac{1}{\sum_{m=1}^{N_\theta}\omega^m} \times \sum_{m=1}^{N_\theta} \omega^m u_t(\theta^m),$$

with the convention $p(y_1|y_{1:0},\theta) = p(y_1|\theta)$ for $t = 1$.

(b) Update the importance weights,

$$\omega^m \leftarrow \omega^m u_t(\theta^m). \tag{6.4}$$

(c) If some degeneracy criterion is fulfilled, sample $\tilde{\theta}^m$ independently from the mixture distribution

$$\frac{1}{\sum_{m=1}^{N_\theta}\omega^m} \sum_{m=1}^{N_\theta} \omega^m K_t(\theta^m, \cdot).$$

Finally, replace the current weighted particle system, by the set of new, unweighted particles:

$$(\theta^m, \omega^m) \leftarrow (\tilde{\theta}^m, 1).$$

[Chopin 04] shows that

$$\frac{\sum_{m=1}^{N_\theta}\omega^m\varphi(\theta^m)}{\sum_{m=1}^{N_\theta}\omega^m}$$

is a consistent and asymptotically (as $N_\theta \to \infty$) normal estimator of the expectations

$$\mathbb{E}\left[\varphi(\theta)|y_{1:t}\right] = \int \varphi(\theta)p(\theta|y_{1:t})\,d\theta,$$

for all appropriately integrable φ. In addition, each L_t, computed in Step (a), is a consistent and asymptotically normal estimator of the likelihood $p(y_t|y_{1:t-1})$.

Step (c) is usually decomposed into a resampling and a mutation step. In the above algorithm the former is done with the multinomial distribution, where particles are selected with probability proportional to ω^m. As mentioned in Section 6.2.1 other resampling schemes may be used instead. The move step is achieved through a Markov kernel K_t which leaves $p(\theta|y_{1:t})$ invariant. In our examples K_t will be a Metropolis-Hastings kernel. A significant advantage of IBIS is that the population of θ-particles can be used to learn features of the target distribution, e.g by computing

$$\widehat{\Sigma} = \frac{1}{\sum_{m=1}^{N_\theta}\omega^m}\sum_{m=1}^{N_\theta}\omega^m\left(\theta^m - \hat{\mu}\right)\left(\theta^m - \hat{\mu}\right)^T, \quad \hat{\mu} = \frac{1}{\sum_{m=1}^{N_\theta}\omega^m}\sum_{m=1}^{N_\theta}\omega^m\theta^m.$$

New particles can be proposed according to a Gaussian random walk $\tilde{\theta}^m|\theta^m \sim N(\theta^m, c\widehat{\Sigma})$, where c is a tuning constant for achieving optimal scaling of the Metropolis-Hastings algorithm, or independently $\tilde{\theta}^m \sim N(\hat{\mu}, \widehat{\Sigma})$ as suggested in [Chopin 02]. A standard degeneracy criterion is ESS $< \gamma N_\theta$, for $\gamma \in (0,1)$, where ESS stands for "effective sample size" and is computed as

$$\text{ESS} = \frac{\left(\sum_{m=1}^{N_\theta}\omega^m\right)^2}{\sum_{m=1}^{N_\theta}\left(\omega^m\right)^2}. \tag{6.5}$$

Theory and practical guidance on the use of this criterion are provided in Sections 6.3.7 and 6.4 respectively.

In the context of state-space models IBIS is a theoretical algorithm since the likelihood increments $p(y_t|y_{1:t-1},\theta)$ (used both in Step 2, and implicitly in the MCMC kernel) are typically intractable. Nevertheless, coupling IBIS with PFs yields a working algorithm as we show in the following section.

6.3 Sequential parameter and state estimation: the SMC^2 algorithm

SMC^2 is a natural amalgamation of IBIS and PF. We first provide the algorithm, we then demonstrate its validity and we close the section by considering various possibilities in its implementation. Again, all the operations involving the index m must be understood as operations performed for all $m \in 1:N_\theta$.

Sample θ^m from $p(\theta)$ and set $\omega^m \leftarrow 1$. Then, at time $t = 1,\ldots,T$,

(a) For each particle θ^m, perform iteration t of the PF described in Section 6.2.1: If $t=1$, sample independently $x_1^{1:N_x,m}$ from ψ_{1,θ^m}, and compute

$$\hat{p}(y_1|\theta^m) = \frac{1}{N_x}\sum_{n=1}^{N_x} w_{1,\theta}(x_1^{n,m});$$

If $t > 1$ sample $(x_t^{1:N_x,m}, a_{t-1}^{1:N_x,m})$ from ψ_{t,θ^m} conditional on the parents $(x_{1:t-1}^{1:N_x,m}, a_{1:t-2}^{1:N_x,m})$ and compute

$$\hat{p}(y_t|y_{1:t-1},\theta^m) = \frac{1}{N_x}\sum_{n=1}^{N_x} w_{t,\theta}(x_{t-1}^{a_{t-1}^{n,m},m}, x_t^{n,m}).$$

(b) Update the importance weights,

$$\omega^m \leftarrow \omega^m \hat{p}(y_t|y_{1:t-1}, \theta^m). \tag{6.6}$$

(c) If some degeneracy criterion is fulfilled, sample $\left(\tilde{\theta}^m, \tilde{x}_{1:t}^{1:N_x,m}, \tilde{a}_{1:t-1}^{1:N_x}\right)$ independently from the mixture distribution

$$\frac{1}{\sum_{m=1}^{N_\theta} \omega^m} \sum_{m=1}^{N_\theta} \omega^m K_t \left\{ \left(\theta^m, x_{1:t}^{1:N_x,m}, a_{1:t-1}^{1:N_x,m}\right), \cdot \right\}$$

where K_t is a PMCMC kernel described in Section 6.3.2. Finally, replace the current weighted particle system by the set of new unweighted particles:

$$(\theta^m, x_{1:t}^{1:N_x,m}, a_{1:t-1}^{1:N_x,m}, \omega^m) \leftarrow (\tilde{\theta}^m, \tilde{x}_{1:t}^{1:N_x,m}, \tilde{a}_{1:t-1}^{1:N_x,m}, 1).$$

The degeneracy criterion in Step (c) will typically be the same as for IBIS, i.e., when the ESS drops below a threshold, where the ESS is computed as in (6.5) and the ω^m's are now obtained in (6.6). We study the stability and the computational cost of the algorithm when applying this criterion in Section 6.3.7.

6.3.1 Formal justification of SMC²

A proper formalisation of the successive importance sampling steps performed by the SMC² algorithm requires extending the sampling space, in order to include all the random variables generated by the algorithm.

At time $t = 1$, the algorithm generates variables θ^m from the prior $p(\theta)$, and for each θ^m, the algorithm generates vectors $x_1^{1:N_x,m}$ of particles, from $\psi_{1,\theta^m}(x_1^{1:N_x})$. Thus, the sampling space is $\Theta \times \mathcal{X}^{N_x}$, and the actual "particles" of the algorithm are N_θ independent and identically distributed copies of the random variable $(\theta, x_1^{1:N_x})$, with density:

$$p(\theta)\psi_{1,\theta}(x_1^{1:N_x}) = p(\theta) \prod_{n=1}^{N_x} q_{1,\theta}(x_1^n).$$

Then, these particles are assigned importance weights corresponding to the incremental weight function $\hat{Z}_1(\theta, x_1^{1:N_x}) = N_x^{-1} \sum_{n=1}^{N_x} w_{1,\theta}(x_1^n)$. This means that, at iteration 1, the target distribution of the algorithm should be defined as:

$$\pi_1(\theta, x_1^{1:N_x}) = p(\theta)\psi_{1,\theta}(x_1^{1:N_x}) \times \frac{\hat{Z}_1(\theta, x_1^{1:N_x})}{p(y_1)},$$

where the normalising constant $p(y_1)$ is easily deduced from the property that $\hat{Z}_1(\theta, x_1^{1:N_x})$ is an unbiased estimator of $p(y_1|\theta)$. To understand the properties of π_1, simple manipulations suffice. Substituting $w_{1,\theta}(x_1^n)$, $\psi_{1,\theta}(x_1^{1:N_x})$ and $\hat{Z}_1(\theta, x_1^{1:N_x})$ with their respective expressions,

$$\begin{aligned}
\pi_1(\theta, x_1^{1:N_x}) &= \frac{p(\theta)}{p(y_1)} \prod_{i=1}^{N_x} q_{1,\theta}(x_1^i) \left\{ \frac{1}{N_x} \sum_{n=1}^{N_x} \frac{\mu_\theta(x_1^n) g_\theta(y_1|x_1^n)}{q_{1,\theta}(x_1^n)} \right\} \\
&= \frac{1}{N_x} \sum_{n=1}^{N_x} \frac{p(\theta)}{p(y_1)} \mu_\theta(x_1^n) g_\theta(y_1|x_1^n) \left\{ \prod_{i=1, i\neq n}^{N_x} q_{1,\theta}(x_1^i) \right\}
\end{aligned}$$

and noting that, for the triplet (θ, x_1, y_1) of random variables,

$$p(\theta)\mu_\theta(x_1)g_\theta(y_1|x_1) = p(\theta, x_1, y_1) = p(y_1)p(\theta|y_1)p(x_1|y_1, \theta)$$

one finally gets that:

$$\pi_1(\theta, x_1^{1:N_x}) \quad = \quad \frac{p(\theta|y_1)}{N_x}\sum_{n=1}^{N_x} p(x_1^n|y_1,\theta)\left\{\prod_{i=1,i\neq n}^{N_x} q_{1,\theta}(x_1^i)\right\}.$$

The following two properties of π_1 are easily deduced from this expression. First, the marginal distribution of θ is $p(\theta|y_1)$. Thus, at iteration 1 the algorithm is properly weighted for any N_x. Second, conditional on θ, π_1 assigns to the vector $x_1^{1:N_x}$ a mixture distribution which with probability $1/N_x$, gives to particle n the filtering distribution $p(x_1|y_1,\theta)$, and to all the remaining particles the proposal distribution $q_{1,\theta}$. The notation reflects these properties by denoting the target distribution of SMC² by π_1, since it admits the distributions defined in (6.1) as marginals.

By a simple induction, one sees that the target density π_t at iteration $t \geq 2$ should be defined as:

$$\pi_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}) = p(\theta)\psi_{t,\theta}(x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}) \times \frac{\hat{Z}_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})}{p(y_{1:t})} \tag{6.7}$$

where $\hat{Z}_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})$ was defined in (6.3), that is, it should be proportional to the sampling density of all random variables generated so far, times the product of the successive incremental weights. Again, the normalising constant $p(y_{1:t})$ in (6.7) is easily deduced from the fact that $\hat{Z}_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})$ is an unbiased estimator of $p(y_{1:t}|\theta)$. The following Proposition gives an alternative expression for π_t.

Proposition 1. *The probability density π_t may be written as:*

$$\pi_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}) = p(\theta|y_{1:t})\times \tag{6.8}$$

$$\frac{1}{N_x}\sum_{n=1}^{N_x}\frac{p(\mathbf{x}_{1:t}^n|\theta,y_{1:t})}{N_x^{t-1}}\left\{\prod_{\substack{i=1\\ i\neq \mathbf{h}_t^n(1)}}^{N_x} q_{1,\theta}(x_1^i)\right\}\left\{\prod_{s=2}^{t}\prod_{\substack{i=1\\ i\neq \mathbf{h}_t^n(s)}}^{N_x} W_{s-1,\theta}^{a_{s-1}^i} q_{s,\theta}(x_s^i|x_{s-1}^{a_{s-1}^i})\right\}$$

where $\mathbf{x}_{1:t}^n$ and $\mathbf{h}_t^n$ are deterministic functions of $x_{1:t}^{1:N_x}$ and $a_{1:t-1}^{1:N_x}$ defined as follows: $\mathbf{h}_t^n = (\mathbf{h}_t^n(1),\ldots,\mathbf{h}_t^n(t))$ denote the index history of x_t^n, that is, $\mathbf{h}_t^n(t) = n$, and $\mathbf{h}_t^n(s) = a_s^{\mathbf{h}_t^n(s+1)}$, recursively, for $s = t-1,\ldots,1$, and

$$\mathbf{x}_{1:t}^n = (\mathbf{x}_{1:t}^n(1),\ldots,\mathbf{x}_{1:t}^n(t))$$

denote the state trajectory of particle x_t^n, i.e. $\mathbf{x}_{1:t}^n(s) = x_s^{\mathbf{h}_t^n(s)}$, for $s = 1,\ldots,t$.

A proof is given in Appendix A. We use a bold notation to stress out that the quantities $\mathbf{x}_{1:t}^n$ and $\mathbf{h}_t^n$ are quite different from particle arrays such as e.g. $x_{1:t}^{1:N_x}$: $\mathbf{x}_{1:t}^n$ and $\mathbf{h}_t^n$ provide the complete genealogy of the particle with label n at time t, while $x_{1:t}^{1:N_x}$ simply concatenates the successive particle arrays $x_t^{1:N_x}$, and contains no such genealogical information.

It follows immediately from expression (6.8) that the marginal distribution of π_t with respect to θ is $p(\theta|y_{1:t})$. Conditional on θ the remaining random variables, $x_{1:t}^{1:N_x}$ and $a_{1:t-1}^{1:N_x}$, have a mixture distribution, according to which, with probability $1/N_x$ the state trajectory $\mathbf{x}_{1:t}^n$ is generated according to $p(x_{1:t}|\theta, y_{1:t})$, the ancestor variables corresponding to this trajectory, $a_s^{\mathbf{h}_t^n(s)}$ are uniformly distributed within $1:N_x$, and all the other random variables are generated from the particle filter proposal distribution, $\psi_{t,\theta}$. Therefore, Proposition 1 establishes a sequence of auxiliary distributions π_t on increasing dimensions, whose marginals include the posterior distributions of interest defined in (6.1). The SMC² algorithm targets this sequence using SMC techniques.

6.3.2 The MCMC rejuvenation step

To formally describe this step performed at some iteration t, we must work, as in the previous section, on the extended set of variables $(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})$. The algorithm is described below; if the proposed move is accepted, the set of variables is replaced by the proposed one, otherwise it is left unchanged. The algorithm is based on some proposal kernel $T(\theta, d\tilde{\theta})$ in the $\theta-$dimension, which admits probability density $T(\theta, \tilde{\theta})$. (The proposal kernel for θ, $T(\theta, \cdot)$, may be chosen as described in Section 6.2.2.)

(a) Sample $\tilde{\theta}$ from proposal kernel, $\tilde{\theta} \sim T(\theta, d\tilde{\theta})$.

(b) Run a new PF for $\tilde{\theta}$:

- sample independently $(\tilde{x}_{1:t}^{1:N_x}, \tilde{a}_{1:t-1}^{1:N_x})$ from $\psi_{t,\tilde{\theta}}$,
- and compute $\hat{Z}_t(\tilde{\theta}, \tilde{x}_{1:t}^{1:N_x}, \tilde{a}_{1:t-1}^{1:N_x})$.

(c) Accept the move with probability

$$1 \wedge \frac{p(\tilde{\theta}) \hat{Z}_t(\tilde{\theta}, \tilde{x}_{1:t}^{1:N_x}, \tilde{a}_{1:t-1}^{1:N_x}) T(\tilde{\theta}, \theta)}{p(\theta) \hat{Z}_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}) T(\theta, \tilde{\theta})} .$$

It directly follows from (6.7) that this algorithm defines a standard Hastings-Metropolis kernel with proposal distribution

$$q_\theta(\tilde{\theta}, \tilde{x}_{1:t}^{1:N_x}, \tilde{a}_{1:t}^{1:N_x}) = T(\theta, \tilde{\theta}) \psi_{t,\tilde{\theta}}(\tilde{x}_{1:t}^{1:N_x}, \tilde{a}_{1:t}^{1:N_x})$$

and admits as invariant distribution the distribution $\pi_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})$.

In the broad PMCMC framework, this scheme corresponds to the so-called particle Metropolis-Hastings algorithm [see Andrieu 10]. It is worth pointing out an interesting digression from the PMCMC framework. The Markov mutation kernel has to be invariant with respect to π_t, but it does not necessarily need to produce an ergodic Markov chain, since consistency of Monte Carlo estimates is achieved by averaging across many particles and not within a path of a single particle. Hence, we can also attempt lower dimensional updates, e.g using a Hastings-within-Gibbs algorithm. The advantage of such moves is that they might lead to higher acceptance rates for the same step size in the θ-dimension. However, we do not pursue this point further in this article.

6.3.3 PMCMC's invariant distribution, state inference

From (6.8), one may rewrite π_t as the marginal distribution of $(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})$ with respect to an extended distribution that would include a uniformly distributed particle index $n^\star \in 1 : N_x$:

$$\pi_t^\star(n^\star, \theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}) = \frac{p(\theta|y_{1:t})}{N_x^t} \times \\ p(\mathbf{x}_{1:t}^{n^\star}|\theta, y_{1:t}) \left\{ \prod_{\substack{i=1 \\ i \neq \mathbf{h}_t^{n^\star}(1)}}^{N_x} q_{1,\theta}(x_1^i) \right\} \left\{ \prod_{s=2}^{t} \prod_{\substack{i=1 \\ i \neq \mathbf{h}_t^{n^\star}(s)}}^{N_x} W_{s-1,\theta}^{a_{s-1}^i} q_{s,\theta}(x_s^i | x_{s-1}^{a_{s-1}^i}) \right\} . \tag{6.9}$$

[Andrieu 10] formalise PMCMC algorithms as MCMC algorithms that leaves $\pi_t^\star$ invariant, whereas in the previous section we justified our PMCMC update as a MCMC step leaving π_t invariant. This distinction is a mere technicality in the PMCMC context, but it becomes important in the sequential context. SMC^2 is best understood as an algorithm targetting the sequence

(π_t): defining importance sampling steps between successive versions of $\pi_t^\star$ seems cumbersome, as the interpretation of $n^\star$ at time t does not carry over to iteration $t+1$. This distinction also relates to the concept of Rao-Blackwellised (marginalised) particle filters [Doucet 00]: since π_t is a marginal distribution with respect to $\pi_t^\star$, targetting π_t rather than $\pi_t^\star$ leads to more efficient (in terms of Monte Carlo variance) SMC algorithms.

The interplay between π_t and $\pi_t^\star$ is exploited below and in the following sections in order to fully realize the implementation potential of SMC^2. As a first example, direct inspection of (6.9) reveals that the conditional distribution of $n^\star$, given θ, $x_{1:t}^{1:N_x}$ and $a_{1:t-1}^{1:N_x}$, is $\mathcal{M}(W_{t,\theta}^{1:N_x})$, the multinomial distribution that assigns probability $W_{t,\theta}^n$ to outcome n, $n \in 1:N_x$. Therefore, weighted samples from $p(\theta, x_{1:t}|y_{1:t})$ may be obtained at iteration t as follows:

(a) For $m = 1, \ldots, N_\theta$, draw index $n^\star(m)$ from $\mathcal{M}(W_{t,\theta^m}^{1:N_x})$.

(b) Return the weighted sample

$$(\omega^m, \theta^m, \mathbf{x}_{1:t}^{n^\star(m),m})_{m \in 1:N_\theta}$$

where $\mathbf{x}_{1:t}^{n,m}$ was defined in Proposition 1.

This *temporarily extended* particle system can be used in the standard way to make inferences about x_t (filtering), y_{t+1} (prediction) or even $x_{1:t}$ (smoothing), under parameter uncertainty. Smoothing requires to store all the state variables $x_{1:t}^{1:N_x,1:N_\theta}$, which is expensive, but filtering and prediction may be performed while storing only the most recent state variables, $x_t^{1:N_x,1:N_\theta}$. We discuss more thoroughy the memory cost of SMC^2, and explain how smoothing may still be carried out at certain times, without storing the complete trajectories, in Section 6.3.7.

The phrase *temporarily extended* in the previous paragraph refers to our discussion on the difference between π_t and $\pi_t^\star$. By extending the particles with a $n^\star$ component, one temporarily change the target distribution, from π_t to $\pi_t^\star$. To propagate to time $t+1$, one must revert back to π_t, by simply marginaling out the particle index $n^\star$. We note however that, before reverting to π_t, one has the liberty to apply MCMC updates with respect to $\pi_t^\star$. For instance, one may update the $\theta-$component of each particle according to the full conditional distribution of θ with respect to to $\pi_t^\star$, that is, $p(\theta|\mathbf{x}_{1:t}^{n^\star}, y_{1:t})$. Of course, this possibility is interesting mostly for those models such that $p(\theta|\mathbf{x}_{1:t}^{n^\star}, y_{1:t})$ is tractable. And, again, this operation may be performed only if all the state variables are available in memory.

6.3.4 Reusing all the $x-$particles

The previous section describes an algorithm for obtaining a particle sample

$$(\omega^m, \theta^m, \mathbf{x}_{1:t}^{n^\star(m),m})_{m \in 1:N_\theta}$$

that targets $p(\theta, x_{1:t}|y_{1:t})$. One may use this sample to compute, for any test function $h(\theta, x_{1:t})$, an estimator of the expectation of h with respect to the target $p(\theta, x_{1:t}|y_{1:t})$:

$$\frac{1}{\sum_{m=1}^{N_\theta} \omega^m} \sum_{m=1}^{N_\theta} \omega^m h(\theta^m, \mathbf{x}_{1:t}^{n^\star(m),m}).$$

As in [Andrieu 10, Section 4.6], we may deduce from this expression a Rao-Blackwellised estimator, by marginalising out $n^\star$, and re-using all the x-particles:

$$\frac{1}{\sum_{m=1}^{N_\theta} \omega^m} \sum_{m=1}^{N_\theta} \omega^m \left\{ \sum_{n=1}^{N_x} W_{t,\theta^m}^n h(\theta^m, \mathbf{x}_{1:t}^{n,m}) \right\}.$$

The variance reduction obtained by this Rao-Blackwellisation scheme should depend on the variability of $h(\theta^m, \mathbf{x}_{1:t}^{n,m})$ with respect to n. For a fixed m, the components $\mathbf{x}_{1:t}^{n,m}(s)$ of the trajectories $\mathbf{x}_{1:t}^{n,m}$ are diverse when s is close to t, and degenerate when s is small. Thus, this Rao-Blackwellisation scheme should be more efficient when h depends mostly on recent state values, e.g. $h(\theta, x_{1:t}) = h(x_t)$, and less efficient when h depends mostly on early state values, e.g. $h(\theta, x_{1:t}) = h(x_1)$.

6.3.5 Evidence

The evidence of the data obtained up to time t may be decomposed using the chain rule:

$$p(y_{1:t}) = \prod_{s=1}^{t} p(y_s|y_{1:s-1}).$$

The IBIS algorithm delivers the weighted averages L_s, for each $s = 1, \ldots, t$, which are Monte Carlo estimates of the corresponding factors in the product; see Section 6.2.2. Thus, it provides an estimate of the evidence by multiplying these terms. This can also be achieved via the SMC^2 algorithm in a similar manner:

$$\hat{L}_t = \frac{1}{\sum_{m=1}^{N_\theta} \omega^m} \sum_{m=1}^{N_\theta} \omega^m \hat{p}(y_t|y_{1:t-1}, \theta^m)$$

where $\hat{p}(y_t|y_{1:t-1}, \theta^m)$ is given in the definition of the algorithm. It is therefore possible to estimate the evidence of the model, at each iteration t, at practically no extra cost.

6.3.6 Automatic calibration of N_x

The plain vanilla SMC^2 algorithm assumes that N_x stays constant during the complete run. This poses two practical difficulties. First, choosing a moderate value of N_x that leads to a good performance (in terms of small Monte Carlo error) is typically difficult, and may require tedious pilot runs. As any tuning parameter, it would be nice to design a strategy that determines automatically a reasonable value of N_x. Second, [Andrieu 10] show that, in order to obtain reasonable acceptance rates for a particle Metropolis-Hastings step, one should take $N_x = \mathcal{O}(t)$, where t is the number of data-points currently considered. In the SMC^2 context, this means that it may make sense to use a small value for N_x for the early iterations, and then to increase it regularly. Finally, when the variance of the PF estimates depends on θ, it might be interesting to allow N_x to change with θ as well.

The SMC^2 framework provides more scope for such adaptation compared to PMCMC. In this section we describe two possibilities, which relate to the two main particle MCMC methods, particle marginal Metropolis-Hastings and particle Gibbs. The former generates the auxiliary variables independently of the current particle system whereas the latter does it conditionally on the current system. For this reason the latter yields a new system without changing the weights, which is a nice feature, but it requires storing particle histories, which is memory inefficient; see Section 6.3.7 for a more thorough discussion of the memory cost of SMC^2.

The schemes for increasing N_x can be integrated into the main SMC^2 algorithm along with rules for automatic calibration. We propose the following simple strategy. We start with a small value for N_x, we monitor the acceptance rate of the PMCMC step and when this rate falls below a given threshold, we trigger the “changing N_x” step; for example we multiply N_x by 2.

Exchange importance sampling step

Our first suggestion involves a particle exchange. At iteration t, the algorithm has generated so far the random variables $\theta^{1:N_\theta}$, $x_{1:t}^{1:N_x, 1:N_\theta}$ and $a_{1:t-1}^{1:N_x, 1:N_\theta}$ and the target distribution is $\pi_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})$. At this stage, one may extend the sampling space, by generating for each particle θ^m, new PFs of size $\tilde{N}_x$, by simply sampling independently, for each m, the random variables $\tilde{x}_{1:t}^{1:\tilde{N}_x,m}$, $\tilde{a}_{1:t-1}^{1:\tilde{N}_x,m}$ from ψ_{t,θ^m}. Thus, the extended target distribution is:

$$\pi_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})\psi_{t,\theta}(\tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x}). \tag{6.10}$$

In order to swap the $x-$particles and the $\tilde{x}-$particles, we use the generalised importance sampling strategy of [Del Moral 06], which is based on an artificial backward kernel. Using (6.7), we compute the incremental weights

$$\begin{aligned} &\frac{\pi_t(\theta, \tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x})L_t\left((\theta, \tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x}), (x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})\right)}{\pi_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})\psi_{t,\theta}(\tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x})} \\ &= \frac{\hat{Z}_t(\theta, \tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x})}{\hat{Z}_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})} \times \frac{L_t\left((\theta, \tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x}), (x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})\right)}{\psi_{t,\theta}(x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})} \end{aligned} \tag{6.11}$$

where L_t is a backward kernel density. One then may drop the "old" particles $(x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})$ in order to obtain a new particle system, based on particles $(\theta, \tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x})$ targetting π_t, but with $\tilde{N}_x$, $x-$particles.

This importance sampling operation is valid under mild assumptions for the backward kernel L_t; namely that the support of the denominator of (6.11) is included in the support of its numerator. One easily deduces from Proposition 1 of [Del Moral 06] and (6.7) that the optimal kernel (in terms of minimising the variance of the weights) is

$$L_t^{\text{opt}}\left((\theta, \tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x}), (x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})\right) = \frac{\psi_{t,\theta}(x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})\hat{Z}_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})}{p(y_{1:t}|\theta)}.$$

This function is intractable, because of the denominator $p(y_{1:t}|\theta)$, but it suggests the following simple approximation: L_t should be set to $\psi_{t,\theta}$, so as to cancel the second ratio, which leads to the very simple incremental weight function:

$$u_t^{exch}\left(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}, \tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x}\right) = \frac{\hat{Z}_t(\theta, \tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x})}{\hat{Z}_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})}.$$

By default, one may implement this exchange step for all the particles $\theta^{1:N_\theta}$, and multiply consequently each particle weight ω^m with the ratio above. However, it is possible to apply this step to only a subset of particles, either selected randomly or according to some deterministic criterion based on θ. (In that case, only the weights of the selected particles should be updated.) Similarly, one could update certain particles according to a Hastings-Metropolis step, where the exchange operation is proposed, and accepted with probabilty the minimum of 1 and the ratio above.

In both cases, one effectively targets a mixture of π_t distributions corresponding to different values of N_x. This does not pose any formal difficulty, because these distributions admit the same marginal distributions with respect to the components of interest (θ, and $x_{1:t}$ if the target distribution is extended as described in Section 6.3.3), and because the successive importance sampling steps (such as either the exchange step above, or Step (b) in the SMC^2 Algorithm) correspond to ratios of densities that are known up to a constant that does not depend on N_x.

Of course, in practice, propagating PF of varying size N_x is a bit more cumbersome to implement, but it may show useful in particular applications, where for instance the computational cost of sampling a new state x_{t+1}, conditional on x_t, varies strongly according to θ.

Conditional SMC step

Whereas the exchange steps associates with the target π_t, and the particle Metropolis-Hastings algorithm, our second suggestion relates to the target $\pi_t^\star$, and to the particle Gibbs algorithm.

First, one extends the target distribution, from π_t to $\pi_t^\star$, by sampling a particle index $n^\star$, as explained in Section 6.3.3. Then one may apply a conditional SMC step [Andrieu 10], to generate a new particle filter of size $\tilde{N}_x$, $\tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x}$, but conditional on one trajectory being equal to $\mathbf{x}_{1:t}^n$. This amounts to sampling the conditional distribution defined by the two factors in curly brackets in (6.9), which can also be conveniently rewritten as

$$\frac{N_x^t \pi_t^\star(n^\star, \theta, \tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x})}{p(\theta, \mathbf{x}_{1:t}^n | y_{1:t})}.$$

We refrain from calling this operation a Gibbs step, because it changes the target distribution (and in particular its dimension), from $\pi_t(\theta, x_{1:t}^{1:N_x}, a_{1:t}^{1:N_x})$ to $\pi_t(\theta, x_{1:t}^{1:\tilde{N}_x}, a_{1:t}^{1:\tilde{N}_x})$. A better formalisation is again in terms of an importance sampling step involving a backward kernel [Del Moral 06], from the proposal distribution, the current target distribution π_t times the conditional distribution of the newly generated variables:

$$\pi_t^\star(n^\star, \theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}) \frac{N_x^t \pi_t^\star(n^\star, \theta, \tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x})}{p(\theta, \mathbf{x}_{1:t}^n | y_{1:t})}$$

towards target distribution

$$\pi_t^\star(n^\star, \theta, \tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x}) L_t\left((n^\star, \theta, \tilde{x}_{1:t}^{1:\tilde{N}_x}, \tilde{a}_{1:t-1}^{1:\tilde{N}_x}), \cdot\right)$$

where L_t is again an arbitrary backward kernel, whose argument, denoted by a dot, is all the variables in $(x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})$, except the variables corresponding to trajectory $\mathbf{x}_{1:t}^n$. It is easy to see that the optimal backward kernel (applying again Proposition 1 of [Del Moral 06]) is such that the importance sampling ratio equals one. The main drawback of this approach is that it requires to store all the state variables $x_{1:t}^{1:N_x, 1:N_\theta}$; see our dicussion of memory cost in Section 6.3.7.

6.3.7 Complexity

Memory cost

In full generality the SMC^2 algorithm is memory-intensive: up to iteration t, $\mathcal{O}(tN_\theta N_x)$ variables have been generated and potentially have to be carried forward to the next iteration. We explain now how this cost can be reduced to $\mathcal{O}(N_\theta N_x)$ with little loss of generality.

Only the variables $x_{t-1}^{1:N_x, 1:N_\theta}$ are necessary to carry out Step (a) of the algorithm, while all other state variables $x_{1:t-2}^{1:N_x, 1:N_\theta}$ can be discarded. Additionally, when Step (c) is carried out as described in Section 6.3.2, $\hat{Z}_t$ is the only additional necessary statistic of the particle histories. Thus, the typical implementation of SMC^2 for sequential parameter estimation, filtering and prediction has an $\mathcal{O}(N_\theta N_x)$ memory cost. The memory cost of the exchange step is also $O(N_x N_\theta)$; more precisely, it is $O(\tilde{N}_x N_\theta)$, where $\tilde{N}_x$ is the new size of the PF's. A nice property of this exchange step is that it temporarily regenerates complete trajectories $x_{1:t}^{1:\tilde{N}_x, m}$, sequentially for $m = 1, \ldots, M$. Thus, besides augmenting N_x dynamically, the exchange step can also be used to to carry out operations involving complete trajectories at certain pre-defined times, while maintaining a $\mathcal{O}(N_\theta \tilde{N}_x)$ overall cost. Such operations include inference with respect to $\pi_t(\theta, x_{1:t})$, updating θ with respect to the full conditional $p(\theta | x_{1:t}, y_{1:t})$, as explained in Section 6.3.3, or even the conditional SMC update descrided in Section 6.3.6.

Stability and computational cost

Step (c), which requires re-estimating the likelihood, is the most computationally expensive component of SMC^2. When this operation is performed at time t, it incurs an $\mathcal{O}(tN_\theta N_x)$ computational cost. Therefore, to study the computational cost of SMC^2 we need to investigate the rate at which ESS drops below a given threshold. This question directly relates to the stability of the filter, and we will work as in Section 3.1 of [Chopin 04] to answer it. Our approach is based on certain simplifying assumptions, regularity conditions and a recent result of [Cérou 11] which all lead to Proposition 2; the assumptions are discussed in some detail in Appendix B.

In general, ESS$/N_\theta < \gamma$, for ESS given in (6.5), is a standard degeneracy criterion of sequential importance sampling due to the fact that the limit of ESS$/N_\theta$ as $N_\theta \to \infty$ is equal to the inverse of the second moment of the importance sampling weights (normalized to have mean 1). This limiting quantity, which we will generically denote by $\mathcal{E}$, is also often called effective sample size since it can be interpreted as an equivalent number of independent samples from the target distribution [see Section 2.5.3 of Liu 08, for details]. The first simplification in our analysis is to study the properties of $\mathcal{E}$, rather than its finite sample estimator ESS$/N_\theta$, and consider an algorithm which resamples whenever $\mathcal{E} < \gamma$.

Consider now the specific context of SMC^2. Let t be a resampling time at which equally weighted, independent particles have been obtained, and $t+p$, $p>0$, a future time such that no resampling has happened since t. The marginal distribution of the resampled particles at time t is only approximately π_t due to the burn-in period of the Markov chains which are used to generate them. The second simplifying assumption in our analysis is that this marginal distribution is precisely π_t. Under this assumption, the particles at time $t+p$ are generated according to the distribution $\bar{\pi}_{t,t+p}$,

$$\bar{\pi}_{t,t+p}(\theta, x_{1:t+p}^{1:N_x}, a_{1:t+p-1}^{1:N_x}) = \pi_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})\, \psi_{t+p,\theta}(x_{1:t+p}^{1:N_x}, a_{1:t+p-1}^{1:N_x} \mid x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})$$

and the expected value of the weights ω_{t+p} obtained from (6.6) is $p(y_{1:t})/p(y_{1:t+p})$. Therefore, the normalized weights are given by

$$\frac{p(y_{1:t})}{p(y_{1:t+p})} \prod_{i=1}^{p} \hat{p}(y_{t+i}|y_{1:t+i-1},\theta) \triangleq \frac{\hat{Z}_{t+p|t}(\theta, x_{1:t+p}^{1:N_x}, a_{1:t+p-1}^{1:N_x})}{p(y_{t+1:t+p}|y_{1:t})},$$

and the inverse of the second moment of the normalized weights in SMC^2 and IBIS is given by

$$\mathcal{E}_{t,t+p}^{N_x} = \left\{ \mathbb{E}_{\bar{\pi}_{t,t+p}} \left[\frac{\hat{Z}_{t+p|t}(\theta, x_{1:t+p}^{1:N_x}, a_{1:t+p-1}^{1:N_x})^2}{p(y_{t+1:t+p}|y_{1:t})^2} \right] \right\}^{-1},$$

$$\mathcal{E}_{t,t+p}^{\infty} = \left\{ \mathbb{E}_{p(\theta|y_{1:t})} \left[\frac{p(\theta|y_{1:t+p})^2}{p(\theta|y_{1:t})^2} \right] \right\}^{-1}.$$

The previous development leads to the following Proposition which is proved in Appendix B.

Proposition 2. *1. Under Assumptions (H1a) and (H1b) in Appendix B, there exists a constant $\eta > 0$ such that for any p, if $N_x > \eta p$,*

$$\mathcal{E}_{t,t+p}^{N_x} \geq \frac{1}{2}\mathcal{E}_{t,t+p}^{\infty}. \tag{6.12}$$

2. Under Assumptions (H2a)-(H2d) in Appendix B, for any $\gamma > 0$ there exist $\tau, \eta > 0$ and $t_0 < \infty$, such that for $t \geq t_0$,

$$\mathcal{E}_{t,t+p}^{N_x} \geq \gamma, \text{ for } p = \lceil \tau t \rceil,\ N_x = \lceil \eta t \rceil.$$

The implication of this Proposition is the following: under the assumptions in Appendix B and the assumption that the resampling step produces samples from the target distribution, the resample steps should be triggered at times $\lceil \tau^k \rceil$, $k \geq 1$, to ensure that the weight degeneracy between two successive resampling step stays bounded in the run of the algorithm; at these times N_x should be adjusted to $N_x = \lceil \eta \tau^k \rceil$; thus, the cost of each successive importance sampling step is $\mathcal{O}(N_\theta \tau^k)$, until the next resampling step; a simple calculation shows that the cumulative computational cost of the algorithm up to some iteration t is then $\mathcal{O}(N_\theta t^2)$. This is to be contrasted with a computational cost $\mathcal{O}(N_\theta t)$ for IBIS under a similar set of assumptions. The assumptions which lead to this result are restrictive but they are typical of the state of the art for obtaining results about the stability of this type of sequential algorithms; see Appendix B for further discussion.

6.4 Numerical illustrations

An initial study which illustrates SMC^2 in a range of examples of moderate difficulty is available from the second author's web-page, see http://sites.google.com/site/pierrejacob/, as supplementary material. In that study, SMC^2 was shown to typically outperform competing algorithms, whether in sequential scenarios (where datapoints are obtained sequentially) or in batch scenarios (where the only distribution of interest is $p(\theta, x_{1:T}|y_{1:T})$ for some fixed time horizon T). For instance, in the former case, SMC^2 was shown to provide smaller Monte Carlo errors than the SOPF at a given CPU cost. In the latter case, SMC^2 was shown to compare favourably to an adaptive version of the marginal PMCMC algorithm proposed by [Peters 10].

In this paper, our objective instead is to take a hammer to SMC^2, that is, to evaluate its performance on models that are regarded as particularly challenging, even for batch estimation purposes. In addition, we treat SMC^2 as much as possible as a black box: the number N_x of x-particles is augmented dynamically (using the exchange step, see Section 6.3.6), as explained in Section 6.3.6; the move steps are calibrated using the current particles, as described at the end of Section 6.2.2, and so on. The only model-dependent inputs are (a) a procedure for sampling from the Markov transition of the model, $f_\theta(x_{t+1}|x_t)$; (b) a procedure for pointwise evaluation the likelihood $g_\theta(y_t|x_t)$; and (c) a prior distribution on the parameters. This means that the proposal $q_{t,\theta}$ is set to the default choice $f_\theta(x_{t+1}|x_t)$. This also means that we are able to treat models such that the density $f_\theta(x_{t+1}|x_t)$ cannot be computed, even if it may be sampled from; this is the case in the first application we consider.

A generic SMC^2 software package written in Python and C by the second author is available at:

http://code.google.com/p/py-smc2/

6.4.1 Sequential prediction of asset price volatility

SMC^2 is particularly well suited to tackle several of the challenges that arise in the probabilistic modelling of financial time series: prediction is of central importance; risk management requires accounting for parameter and model uncertainty; non-linear models are necessary to capture the features in the data; the length of typical time series is large when modelling medium/low frequency data and vast when considering high frequency observations.

We illustrate some of these possibilities in the context of prediction of daily volatility of asset prices. There is a vast literature on stochastic volatility (SV) models; we simply refer to the excellent exposition in [Barndorff-Nielsen 02] for references, perspectives and second-order properties. The generic framework for daily volatility is as follows. Let s_t be the value of a given financial asset (e.g a stock price or an exchange rate) on the t-th day, and $y_t = 10^{5/2}\log(s_t/s_{t-1})$ be the so-called log-returns (the scaling is done for numerical convenience). The SV model specifies a state-space model with observation equation:

$$y_t = \mu + \beta v_t + v_t^{1/2}\epsilon_t\,, t \geq 1 \tag{6.13}$$

where the ϵ_t is a sequence of independent errors which are assumed to be standard Gaussian. The process v_t is known as the actual volatility and it is treated as a stationary stochastic process. This implies that log-returns are stationary with mixed Gaussian marginal distribution. The coefficient β has both a financial interpretation, as a risk premium for excess volatility, and a statistical one, since for $\beta \neq 0$ the marginal density of log-returns is skewed.

We consider the class of Lévy driven Stochastic Volatility models which were first introduced in [Barndorff-Nielsen 01] and have been intensively studied in the last decade from both the mathematical finance and the statistical community. This family of models is specified via a continuous-time model for the joint evolution of log-price and spot (instantaneous) volatility, which are driven by Brownian motion and Lévy process respectively. The actual volatility is the integral of the spot volatility over daily intervals, and the continuous-time model translates into a state-space model for y_t and v_t as we show below. Details can be found in Sections 2 (for the continuous-time specification) and 5 (for the state-space representation) of the original article. Likelihood-based inference for this class of models is recognized as a very challenging problem, and it has been undertaken among others in [Roberts 04, Griffin 06] and most recently in [Andrieu 10] using PMCMC. On the other hand, [Barndorff-Nielsen 02] develop quasi-likelihood methods using the Kalman filter based on an approximate state-space formulation suggested by the second-order properties of the (y_t, v_t) process.

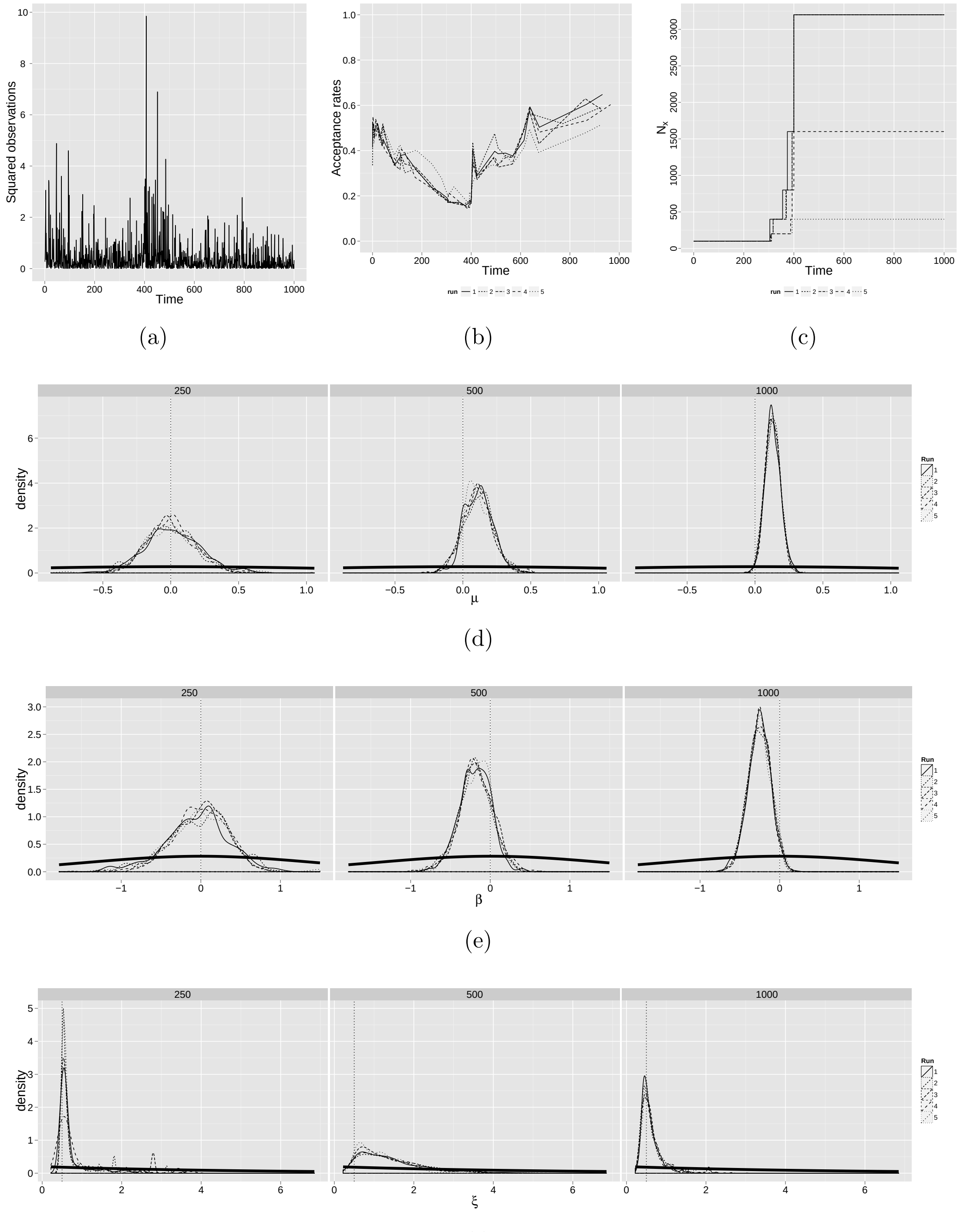
Here we focus on models where the background driving Lévy process is expressed in terms of a finite rate Poisson process and consider multi-factor specifications of such models which include leverage. This choice allows the exact simulation of the actual volatility process, and permits direct comparisons to the numerical results in Sections 4 of [Roberts 04], 3.2 of [Barndorff-Nielsen 02] and 6 of [Griffin 06]. Additionally, this case is representative of a system which can be very easily simulated forwards whereas computation of its transition density is considerably involved (see (6.14) below). The specification for the one-factor model is as follows. We parametrize the latent process as in [Barndorff-Nielsen 02] in terms of (ξ, ω^2, λ) where ξ and ω^2 are the stationary mean and variance of the spot volatility process, and λ the exponential rate of decay of its autocorrelation function. The second-order properties of v_t can be expressed as functions of these parameters, see Section 2.2 of [Barndorff-Nielsen 02]. The state dynamics for the actual volatility are as follows:

$$\begin{gathered} k \sim \text{Poi}\left(\lambda\xi^2/\omega^2\right), \quad c_{1:k} \overset{iid}{\sim} \text{U}(t, t+1), \quad e_{1:k} \overset{iid}{\sim} \text{Exp}\left(\xi/\omega^2\right), \\ z_{t+1} = e^{-\lambda} z_t + \sum_{j=1}^{k} e^{-\lambda(t+1-c_j)} e_j, \quad v_{t+1} = \frac{1}{\lambda}\left[z_t - z_{t+1} + \sum_{j=1}^{k} e_j\right], \quad x_{t+1} = (v_{t+1}, z_{t+1})'. \end{gathered} \tag{6.14}$$

In this representation, z_t is the discretely-sampled spot volatility process, and the Markovian representation of the state process involves the pair (v_t, z_t). The random variables $(k, c_{1:k}, e_{1:k})$ are generated independently for each time period, and $1 : k$ is understood as the empty set when $k = 0$. These system dynamics imply a $\Gamma(\xi^2/\omega^2, \xi/\omega^2)$ as stationary distribution for z_t. Therefore, we take this to be the initial distribution for z_0.

We applied the algorithm to a synthetic data set of length $T = 1,000$ (Figure 6.1(a)) simulated with the values $\mu = 0$, $\beta = 0$, $\xi = 0.5$, $\omega^2 = 0.0625$, $\lambda = 0.01$ which were used also in the simulation study of [Barndorff-Nielsen 02]. We launched 5 independent runs using $N_\theta = 1,000$, a ESS threshold set at 50%, and the independent Hastings-Metropolis scheme described in Section 6.2.2. The number N_x was set initially to 100, and increased whenever the acceptance rate went below 20% (Figure 6.1(b)-(c)). Figure 6.1(d)-(e) shows estimates of the posterior marginal distribution of some parameters. Note the impact the large jump in the volatility has on N_x, which is systematically (across runs) increased around time 400, and the posterior distribution of the parameters of the volatility process, see Figure 6.1(f).

It is interesting to compare the numerical performance of SMC^2 to that of the SOPF and [Liu 01]'s particle filter (referred to as L&W in the following) for this model and data, and for a comparable CPU budget. The SOPF, if run with $N = 10^5$ particles, collapses to one single particle at about $t = 700$ and is thus completely unusable in this context. L&W is a version of SOPF where the θ-components of the particles are diversified using a Gaussian move that leaves the first two empirical moments of the particle sample unchanged. This move unfortunately introduces a bias which is hard to quantity. We implemented L&W with $N = 2 \times 10^5$ (x, θ)-particles and we set the smoothing parameter h to 10^{-1}; see the Supplement for results with various values of h. This number of particles was to chosen to make the computing time of SMC^2



(a) (b) (c)

(d)

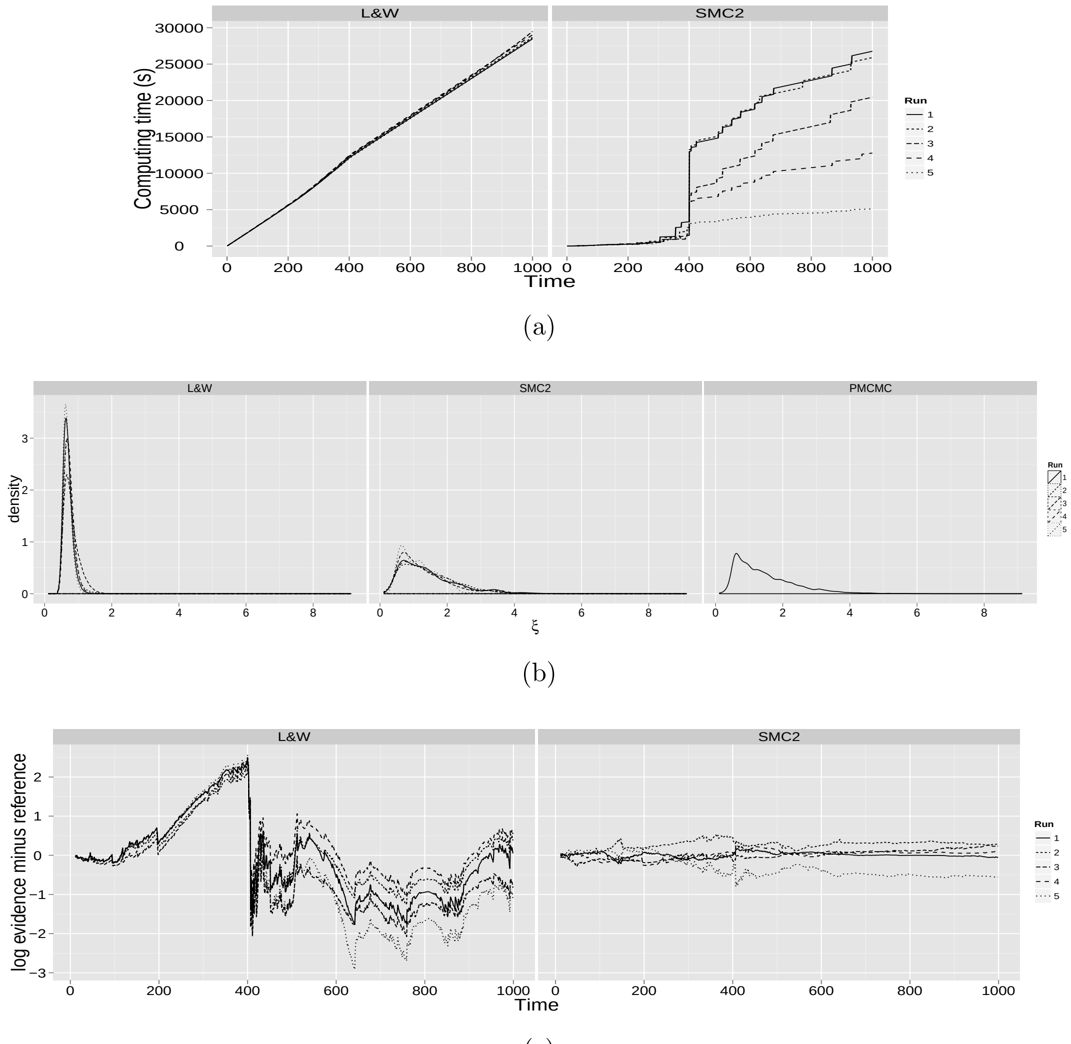
(e)

(f)

Figure 6.1: Single-factor stochastic volatility model, synthetic dataset. (a) Squared observations vs time. (b)-(f) Results obtained from 5 repeated runs: (b) acceptance rate; (c) N_x vs time; (d) to (f) overlaid kernel density estimators of the posterior distribution of μ, β, ξ at different times $t = 250, 500, 1000$, the vertical dashed line indicates the true value and solid red lines the prior density.

and L&W comparable, see Figure 6.2(a). Unsurprisingly, L&W runs are very consistent in terms of computing times, whereas those of SMC^2 are more variable, mainly because the number of x-particles does not reach the same value across the runs and the number of resample-move steps varies. Each of these runs took between 1.5 and 7 hours using a simple Python script and only one processing unit of a 2008 desktop computer (equipped with an Intel Core 2 Duo E8400). Note that, given that these methods could easily be parallelized, the computational cost can be greatly reduced; a 100× speed-up is plausible using appropriate hardware.

Our results suggest that the bias in L&W is significant. Figure 6.2(b) shows the posterior distribution of ξ, the mean of volatility, at time $t = 500$, which is about 100 time steps after the large jump in volatility at time $t = 407$. The results for both algorithms are compared to those from a long PMCMC run [implemented as in Peters 10, and detailed in the Supplement] with $N_x = 500$ and 10^5 iterations. Figure 6.2(c) reports on the estimation of the log evidence $\log p(y_{1:t})$ for each algorithm, plotting the estimated log evidence of each run minus the mean of the log evidence of the 5 SMC^2 runs. We see that the log evidence estimated using L&W is systematically biased, positively or negatively depending on the time steps, with a large discontinuity at time $t = 407$, which is due to underestimation of the tails of the predictive distribution.



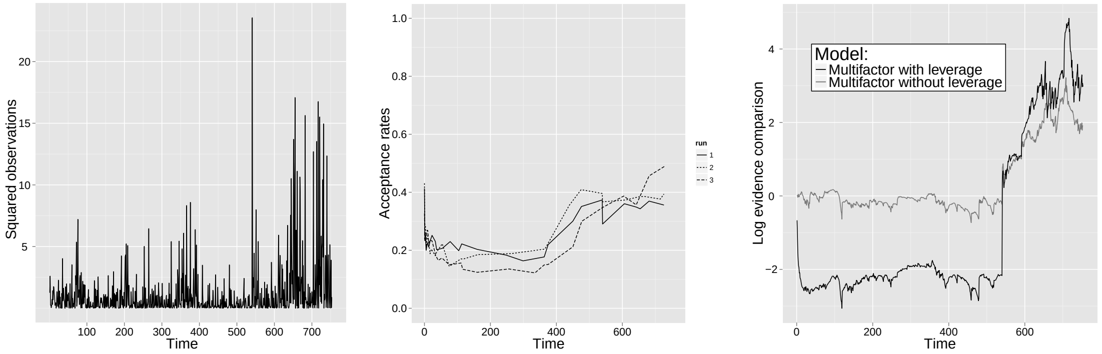
(a)

(b)

(c)

Figure 6.2: Single-factor stochastic volatility model, synthetic dataset, comparison between methods. (a) Computing time of 5 independent runs of L&W (left) and SMC^2 (right) in seconds, against time. (b) Estimation of the posterior marginal distribution of mean volatility, ξ. (c) Estimation of the log evidence, the curves represent the estimated evidence of each run minus the mean across 5 runs of the log evidence estimated using SMC^2.

We now consider models of different complexity for the S&P 500 index. The data set is made of 753 observations from January 3rd 2005 to December 31st 2007 and it is shown on Figure 6.3(a). We first consider a two-factor model, according to which the actual volatility is a sum of two independent components each of which follows a Lévy driven model. Previous research indicates that a two-factor model is sufficiently flexible, whereas more factors do not add significantly when considering daily data, see for example [Barndorff-Nielsen 02, Griffin 06] for Lévy driven models and [Chernov 03] for diffusion-driven SV models. We consider one component which describes long-term movements in the volatility, with memory parameter λ_1, and another which captures short-term variation, with parameter $\lambda_2 >> \lambda_1$. The second component allows more freedom in modelling the tails of the distribution of log-returns. The contribution of the slowly mixing process to the overall mean and variance of the spot volatility is controlled by the parameter $w \in (0, 1)$. Thus, for this model $x_t = (v_{1,t}, z_{1,t}, v_{2,t}, z_{2,t})$ with $v_t = v_{1,t} + v_{2,t}$, where each pair $(v_{i,t}, z_{i,t})$ evolves according to (6.14) with parameters $(w_i\xi, w_i\omega^2, \lambda_i)$ with $w_1 = w, w_2 = 1 - w$. The system errors are generated by independent sets of variables $(k_i, c_{i,1:k}, e_{i,1:k})$, and $z_{0,i}$ are initialized according to the corresponding gamma distributions. Finally, we extend the observation equation to capture a significant feature observed in returns on stocks: low returns provoke increase in subsequent volatility, see for example [Black 76] for an early reference. In parameter driven SV models, one generic strategy to incorporate such feedback is to correlate the noise in the observation and state processes, see [Harvey 96] in the context of the logarithmic SV model, and Section 3 of [Barndorff-Nielsen 01] for Lévy driven models. We take up their suggestion, and re-write the observation equation as



(a) (b) (c)

Figure 6.3: (a): the squares of the S&P 500 data from 03/01/2005 to 21/12/2007. (b): acceptance rates of the resample-move step for the full model over two independent runs. (c): log-evidence comparison between models (relative to the one-factor model).

$$y_t = \mu + \beta v_t + v_t^{1/2}\epsilon_t + \rho_1 \sum_{j=1}^{k_1} e_{1,j} + \rho_2 \sum_{j=1}^{k_2} e_{2,j} - \xi(w\rho_1\lambda_1 + (1-w)\rho_2\lambda_2) \tag{6.15}$$

where $e_{i,j}$ are the system error variables involved in the generation of v_t and ρ_i are the leverage parameters which we expect to be negative. Thus, in this specification we deal with a model with a 5-dimensional state and 9 parameters.

The mathematical tractability of this family of models and the specification in terms of stationary and memory parameters allows to a certain extent subjective Bayesian modelling. Nevertheless, since the main emphasis here is to evaluate the performance of SMC^2 we choose priors that (as we verify a posteriori) are rather flat in the areas of high posterior density. Note that the prior for ξ and ω^2 has to reflect the scaling of the log-returns by a multiplicative factor. We take an Exp(1) prior for λ_1, an Exp(0.5) for $\lambda_2 - \lambda_1$, thus imposing the identifiability constraint $\lambda_2 > \lambda_1$. We take a U(0, 1) prior for w, an Exp(1/5) for ξ and ω^2, and Gaussian priors with large variances for the observation equation parameters.

We launch the three models for the S&P 500 data: single factor, multifactor without and with leverage; note that multifactor without leverage means the full model, but with $\rho_1 = \rho_2 = 0$ in (6.15). We use $N_\theta = 2000$, and N_x is set initially to 100 and then dynamically increases as already described. The acceptance rates stay reasonable as illustrated on Figure 6.3(b). Figure 6.3(c) shows the log evidence $\log p(y_{1:t})$ for the two factor models minus the log evidence for the single factor model. Negative values at time t means that the observations favour the single factor model up to time t. Notice how the model evidence changes after the big jump in volatility around time $t = 550$. Estimated posterior densities for all parameters are provided in the Supplement.

6.4.2 Assessing extreme athletic records

The second application illustrates the potential of SMC^2 in smoothing while accounting for parameter uncertainty. In particular, we consider state-space models that have been proposed for the dynamic evolution of athletic records, see for example [Robinson 95], [Gaetan 04], [Fearnhead 10b]. We analyse the time series of the best times recorded for women's 3000 metres running events between 1976 and 2010. The motivation is to assess to which extent Wang Junxia's world record in 1993 was unusual: 486.11 seconds while the previous record was 502.62 seconds. The data is shown in Figure 6.4(a) and include two observations per year $y = y_{1:2}$, with $y_1 < y_2$: y_1 is the best annual time and y_2 the second best time on the race where y_1 was recorded. The data is available from http://www.alltime-athletics.com/ and it is further discussed in the aforementioned articles. A further fact that sheds doubt on the record is that the second time for 1993 corresponds to an athlete from the same team as the record holder.

We use the same modelling as [Fearnhead 10b]. The observations follow a generalized extreme value (GEV) distribution for minima, with cumulative distribution function G defined by:

$$G(y|\mu, \xi, \sigma) = 1 - \exp\left[-\left\{1 - \xi\left(\frac{y-\mu}{\sigma}\right)\right\}_+^{-1/\xi}\right] \tag{6.16}$$

where μ, ξ and σ are respectively the location, shape and scale parameters, and $\{\cdot\}_+ = \max(0, \cdot)$. We denote by g the associated probability density function. The support of this distribution depends on the parameters; e.g. if $\xi < 0$, g and G are non-zero for $y > \mu + \sigma/\xi$. The probability density function for $y = y_{1:2}$ is given by:

$$g(y_{1:2}|\mu, \xi, \sigma) = \{1 - G(y_2|\mu, \xi, \sigma)\} \prod_{i=1}^{2} \frac{g(y_i|\mu, \xi, \sigma)}{1 - G(y_i|\mu, \xi, \sigma)} \tag{6.17}$$

subject to $y_1 < y_2$. The location μ is not treated as a parameter but as a smooth second-order random walk process:

$$x_t = (\mu_t, \dot{\mu}_t)', \quad x_{t+1} \mid x_t, \nu \sim \mathcal{N}(Fx_t, Q), \quad F = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \text{ and } Q = \nu^2 \begin{pmatrix} 1/3 & 1/2 \\ 1/2 & 1 \end{pmatrix} \tag{6.18}$$

To complete the model specification we set a diffuse initial distribution $\mathcal{N}(520, 10^2)$ on μ_0. Thus we deal with bivariate observations in time $y_t = y_{t,1:2}$, a state-space model with non-Gaussian observation density given in (6.17), a two-dimensional state process given in (6.18), and a 3-dimensional unknown parameter vector, $\theta = (\nu, \xi, \sigma)$. We choose independent exponential prior distributions on ν and σ with rate 0.2. The sign of ξ has determining impact on the support of the observation density, and the computation of extremal probabilities. For this application, given the form of (6.16) and the fact that the observations are necessarily bounded from below, it makes sense to assume that $\xi \leq 0$, hence we take an exponential prior distribution on $-\xi$ with rate 0.5. (We also tried a $N(0, 3^2)$ prior, which had some moderate impact on the estimates presented below, but the corresponding results are not reported here.)

The data we will use in the analysis exclude the two times recorded on 1993. Thus, in an abuse of notation $y_{1976:2010}$ below refers to the pairs of times for all years but 1993, and in the model we assume that there was no observation for that year. Formally we want to estimate probabilities

$$p_t^y = \mathbb{P}(y_t \leq y | y_{1976:2010}) = \int_\Theta \int_\mathcal{X} G(y|\mu_t, \theta) p(\mu_t | y_{1976:2010}, \theta) p(\theta | y_{1976:2010})\, d\mu_t d\theta$$

where the smoothing distribution $p(\mu_t|y_{1976:2010}, \theta)$ and the posterior distribution $p(\theta|y_{1976:2010})$ appear explicitly; below we also consider the probabilities conditionally on the parameter values, rather than integrating over those. The interest lies in $p_{1993}^{486.11}$, $p_{1993}^{502.62}$ and $p_t^{cond} := p_t^{486.11}/p_t^{502.62}$, which is the probability of observing at year t Wang Junxia's record given that we observe a better time than the previous world record. The rationale for using this conditional probability is to take into account the exceptional nature of any new world record.

The algorithm is launched 10 times with $N_\theta = 1,000$ and $N_x = 250$. The resample-move steps are triggered when the ESS goes below 50%, as in the previous example, and the proposal distribution used in the move steps is an independent Gaussian distribution fitted on the particles. The computing time of each of the 10 runs varies between 30 and 70 seconds (using the same machine as in the previous section), which is why we allowed ourselves to use a fairly large number of particles compared to the small time horizon. Figure 6.4(b) represents the estimates $\hat{p}_t^y$ at each year, for $y = 486.11$ (lower box-plots) and $y = 502.62$ (upper box-plots), as well as $\hat{p}_t^{cond} = \hat{p}_t^{486.11}/\hat{p}_t^{502.62}$ (middle box-plots). The box-plots show the variability across the independent runs of the algorithm, and the lines connect the mean values computed across independent runs at each year. The mean value of $\hat{p}_{1993}^{cond}$ over the runs is $9.4 \cdot 10^{-4}$ and the standard deviation over the runs is $3.3 \cdot 10^{-4}$. Note that the estimates $\hat{p}_t^y$ are computed using the smoothing algorithm described in Section 6.3.3.

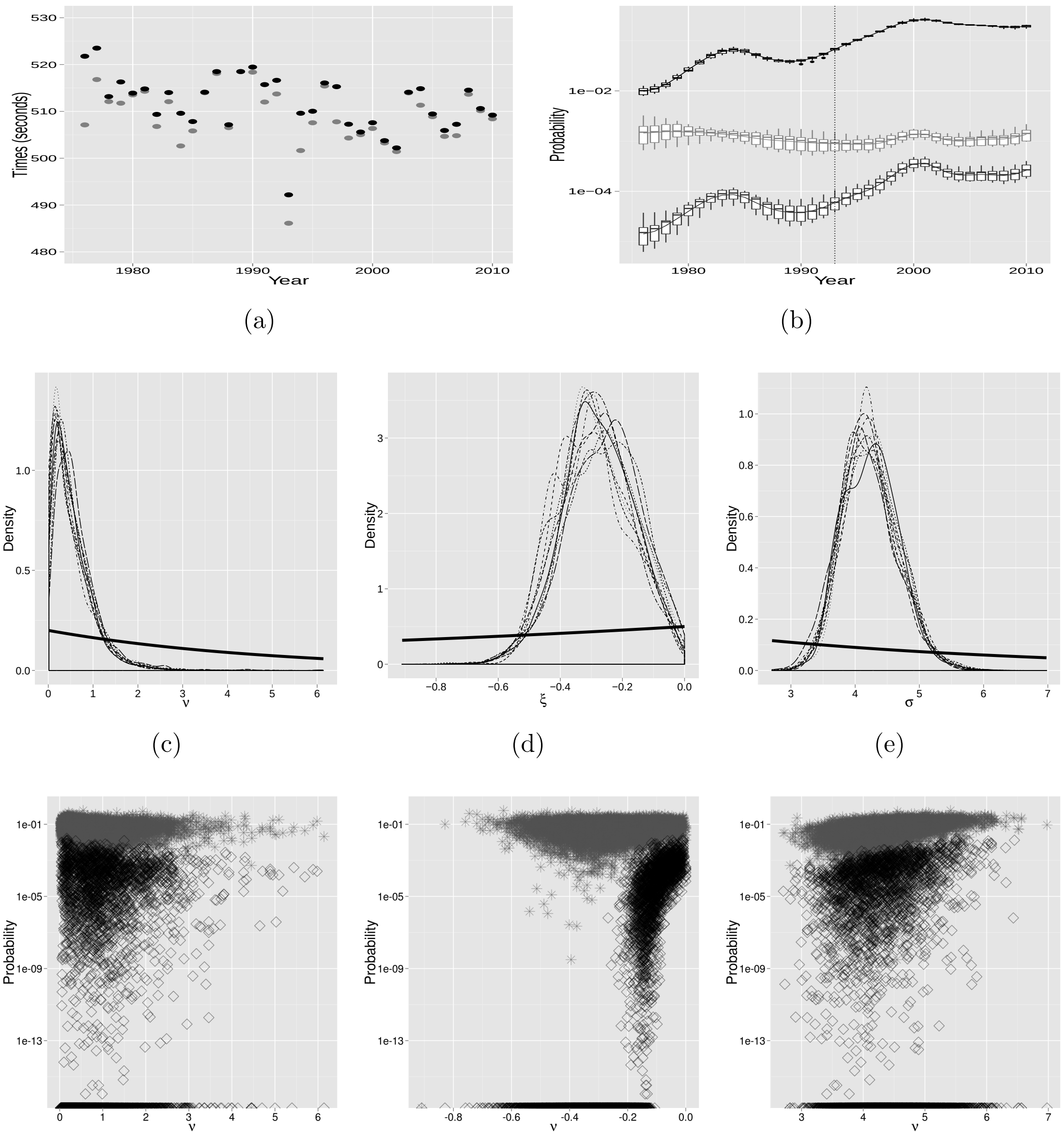
The second row of Figure 6.4 shows the posterior distributions of the three parameters (ν, ξ, σ) using kernel density estimations of the weighted θ-particles. The density estimators obtained for each run are overlaid to show the consistency of the results over independent runs. The prior density function (full line) is nearly flat over the region of high posterior mass. The third row of Figure 6.4 shows scatter plots of the probabilities $G(y|\mu_{1993}^{n^\star(m)}, \theta^m)$ against the parameters θ^m. The triangles represent these probabilities for $y = 486.11$ while the circles represent the probabilities for $y = 502.62$. The cloud of points at the bottom of these plots correspond to parameters θ^m for which the probability $G(486.11|\mu_{1993}^{n^\star(m)}, \theta^m)$ is exactly 0.

6.5 Extensions

In this paper, we developed an "exact approximation" of the IBIS algorithm, that is, an ideal SMC algorithm targetting the sequence $\pi_t(\theta) = p(\theta|y_{1:t})$, with incremental weight $\pi_t(\theta)/\pi_{t-1}(\theta) = p(y_t|y_{1:t-1}, \theta)$. The phrase "exact approximation", borrowed from [Andrieu 10], refers to the fact that our approach targets the exact marginal distributions, for any fixed value N_x.

6.5.1 Intractable densities

We have argued that SMC^2 can cope with state-space models with intractable transition densities provided these can be simulated from. More generally, it can cope with intractable transition of observation densities provided they can be unbiasedly estimated. Filtering for dynamic models with intractable densities for which unbiased estimators can be computed was discussed in [Fearnhead 08]. It was shown that replacing these densities by their unbiased estimators is equivalent to introducing additional auxiliary variables in the state-space model. SMC^2 can directly be applied in this context by replacing these terms by the unbiased estimators to obtain sequential state and parameter inference for such models.



(a) (b)

(c) (d) (e)

(f) (g) (h)

Figure 6.4: Athletics records. (a) Best two times of each year, in women's 3000 metres events between 1976 and 2010; (b) box-plots over 10 runs of SMC^2 of estimates of the probability of interest (top) $p_t^{502.62}$, (middle) p_t^{cond} and (bottom) $p_t^{486.11}$; the y-axis is in log scale and the dotted line indicates the year 1993; (c)-(e) posterior distribution of the parameters approximated by SMC^2, with results of 10 independent runs overlaid on each plot and where the full line represents the prior density function; (f)-(h) probability of observing at year 1993 a recorded time less than 486.11 seconds (blue triangles, lower cloud of points) and less than 502.62 seconds (red circles, upper cloud of points) against the components of θ, where point sizes and transparencies are proportional to the weights of the θ-particles.

6.5.2 SMC^2 for tempering

A natural question is whether we can construct other types of SMC^2 algorithms, which would be "exact approximations" of different SMC strategies. Consider for instance, again for a state-space model, the following geometric bridge sequence [in the spirit of e.g. Neal 01], which allows for a smooth transition from the prior to the posterior:

$$\pi_t(\theta) \propto p(\theta) \left\{p(y_{1:T}|\theta)\right\}^{\gamma_t}, \quad \gamma_t = t/L,$$

where L is the total number of iterations. As pointed out by one referee, see also [Fulop 11a], it is possible to derive some sort of SMC^2 algorithm that targets iteratively the sequence

$$\pi_t(\theta) \propto p(\theta) \left\{\hat{p}(y_{1:T}|\theta)\right\}^{\gamma_t}, \quad \gamma_t = t/L,$$

where $\hat{p}(y_{1:T}|\theta)$ is a particle filtering estimate of the likelihood. Note that $\{\hat{p}(y_{1:T}|\theta)\}^{\gamma_t}$ is not an unbiased estimate of $\{p(y_{1:T}|\theta)\}^{\gamma_t}$ when $\gamma_t \in (0,1)$. This makes the interpretation of the algorithm more difficult, as it cannot be analysed as a noisy, unbiased, version of an ideal algorithm. In particular, Proposition 2 on the complexity of SMC^2 cannot be easily extended to the tempering case. It is also less flexible in terms of PMCMC steps: for instance, it is not possible to implement the conditional SMC step described in Section 6.3.6, or more generally a particle Gibbs step, because such steps rely on the mixture representation of the target distribution, where the mixture index is some selected trajectory, see (6.9), and this representation does not hold in the tempering case. More importantly, this tempering strategy does not make it possible to perform sequential analysis as the SMC^2 algorithm discussed in this paper.

The fact remains that this tempering strategy may prove useful in certain non-sequential scenarios, as suggested by the numerical examples of [Fulop 11a]. It may be used also for determining MAP (maximum a posteriori) estimators, and in particular the maximum likelihood estimator (using a flat prior), by letting $\gamma_t \to +\infty$.

Acknowledgements

N. Chopin is supported by the ANR grant ANR-008-BLAN-0218 "BigMC" of the French Ministry of research. P.E. Jacob is supported by a PhD fellowship from the AXA Research Fund. O. Papaspiliopoulos would like to acknowledge financial support by the Spanish government through a "Ramon y Cajal" fellowship and grant MTM2009-09063. The authors are thankful to Arnaud Doucet (University of Oxford), Peter Müller (UT Austin), Gareth W. Peters (UCL) and the referees for useful comments.

Bibliography

[Andrieu 09] C. Andrieu & G.O. Roberts. *The pseudo-marginal approach for efficient Monte Carlo computations.* The Annals of Statistics, vol. 37, no. 2, pages 697–725, 2009.

[Andrieu 10] C. Andrieu, A. Doucet & R. Holenstein. *Particle Markov chain Monte Carlo methods.* J. R. Statist. Soc. B, vol. 72, no. 3, pages 269–342, 2010.

[Barndorff-Nielsen 01] Ole E. Barndorff-Nielsen & Neil Shephard. *Non-Gaussian Ornstein-Uhlenbeck-based models and some of their uses in financial economics.* J. R. Stat. Soc. Ser. B Stat. Methodol., vol. 63, no. 2, pages 167–241, 2001.

[Barndorff-Nielsen 02] Ole E. Barndorff-Nielsen & Neil Shephard. *Econometric analysis of realized volatility and its use in estimating stochastic volatility models.* J. R. Stat. Soc. Ser. B Stat. Methodol., vol. 64, no. 2, pages 253–280, 2002.

[Black 76] F. Black. *Studies of stock price volatility changes.* In Proceedings of the 1976 meetings of the business and economic statistics section, American Statistical Association, volume 177, page 81, 1976.

[Cappé 04] O Cappé, A. Guillin, J. M Marin & C.P. Robert. *Population Monte Carlo.* J. Comput. Graph. Statist., vol. 23, pages 907–929, 2004.

[Cappé 05] O. Cappé, E. Moulines & T. Rydén. Inference in hidden Markov models. Springer-Verlag, New York, 2005.

[Carpenter 99] J. Carpenter, P. Clifford & P. Fearnhead. *Improved Particle Filter for nonlinear problems.* IEE Proc. Radar, Sonar Navigation, vol. 146, no. 1, pages 2–7, 1999.

[Carvalho 10] C.M. Carvalho, M. Johannes, H.F. Lopes & N. Polson. *Particle learning and smoothing.* Statistical Science, vol. 25, no. 1, pages 88–106, 2010.

[Cérou 11] F. Cérou, P. Del Moral & A. Guyader. *A nonasymptotic theorem for unnormalized Feynman–Kac particle models.* Ann. Inst. Henri Poincarré, vol. 47, no. 3, pages 629–649, 2011.

[Chernov 03] M. Chernov, A. Ronald Gallant, E. Ghysels & G. Tauchen. *Alternative models for stock price dynamics.* Journal of Econometrics, vol. 116, no. 1-2, pages 225–257, 2003.

[Chopin 02] N. Chopin. *A sequential particle filter for static models.* Biometrika, vol. 89, pages 539–552, 2002.

[Chopin 04] N. Chopin. *Central Limit Theorem for sequential Monte Carlo methods and its application to Bayesian inference.* Ann. Stat., vol. 32, no. 6, pages 2385–2411, 2004.

[Chopin 07] N. Chopin. *Inference and model choice for sequentially ordered hidden Markov models.* J. R. Statist. Soc. B, vol. 69, no. 2, pages 269–284, 2007.

[Crisan 02] D. Crisan & A. Doucet. *A survey of convergence results on particle filtering methods for practitioners.* IEEE J. Sig. Proc., vol. 50, no. 3, pages 736–746, 2002.

[Del Moral 99] P. Del Moral & A. Guionnet. *Central limit theorem for nonlinear filtering and interacting particle systems.* Ann. Appl. Prob., vol. 9, pages 275–297, 1999.

[Del Moral 04] P Del Moral. Feynman-kac formulae. Springer, 2004.

[Del Moral 06] P. Del Moral, A. Doucet & A. Jasra. *Sequential Monte Carlo samplers.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 68, no. 3, pages 411–436, 2006.

[Douc 08] R. Douc & E. Moulines. *Limit theorems for weighted samples with applications to sequential Monte Carlo methods.* Ann. Statist., vol. 36, no. 5, pages 2344–2376, 2008.

[Doucet 00] A. Doucet, S. Godsill & C. Andrieu. *On Sequential Monte Carlo Sampling Methods for Bayesian Filtering.* Statist. Comput., vol. 10, no. 3, pages 197–208, 2000.

[Doucet 01] A. Doucet, N. de Freitas & N. J. Gordon. Sequential Monte Carlo methods in practice. Springer-Verlag, New York, 2001.

[Doucet 09] A. Doucet, N. Kantas, S.S. Singh & J.M. Maciejowski. *An Overview of Sequential Monte Carlo Methods for Parameter Estimation in General State-Space Models.* In Proceedings IFAC System Identification (SySid) Meeting., 2009.

[Doucet 11] A. Doucet, G. Poyiadjis & S. Singh. *Sequential Monte Carlo computation of the score and observed information matrix in state-space models with application to parameter estimation.* Biometrika, vol. 98, pages 65–80, 2011.

[Fearnhead 02] P. Fearnhead. *MCMC, Sufficient Statistics and Particle Filters.* Statist. Comput., vol. 11, pages 848–862, 2002.

[Fearnhead 08] P. Fearnhead, O. Papaspiliopoulos & G. O. Roberts. *Particle filters for partially observed diffusions.* J. R. Statist. Soc. B, vol. 70, pages 755–777, 2008.

[Fearnhead 10a] P. Fearnhead, O. Papaspiliopoulos, G.O. Roberts & A. Stuart. *Random weight particle filtering of continuous time processes.* J. R. Stat. Soc. Ser. B Stat. Methodol., vol. 72, pages 497–513, 2010.

[Fearnhead 10b] P. Fearnhead, D. Wyncoll & J. Tawn. *A sequential smoothing algorithm with linear computational cost.* Biometrika, vol. 97, no. 2, page 447, 2010.

[Fulop 11a] A. Fulop & J.C. Duan. *Marginalized Sequential Monte Carlo Samplers.* Rapport technique, SSRN 1837772, 2011.

[Fulop 11b] A. Fulop & J. Li. *Robust and Efficient Learning: A Marginalized Resample-Move Approach.* Rapport technique, SSRN 1724203, 2011.

[Gaetan 04] C. Gaetan & M. Grigoletto. *Smoothing sample extremes with dynamic models.* Extremes, vol. 7, no. 3, pages 221–236, 2004.

[Gilks 01] W. R. Gilks & C. Berzuini. *Following a moving target - Monte Carlo inference for dynamic Bayesian models.* J. R. Statist. Soc. B, vol. 63, pages 127–146, 2001.

[Gordon 93] N. J. Gordon, D. J. Salmond & A. F. M. Smith. *Novel approach to nonlinear/non-Gaussian Bayesian state estimation.* IEE Proc. F, Comm., Radar, Signal Proc., vol. 140, no. 2, pages 107–113, 1993.

[Griffin 06] J.E. Griffin & M.F.J. Steel. *Inference with non-Gaussian Ornstein-Uhlenbeck processes for stochastic volatility.* Journal of Econometrics, vol. 134, no. 2, pages 605–644, 2006.

[Harvey 96] A.C. Harvey & N. Shephard. *Estimation of an asymmetric stochastic volatility model for asset returns.* Journal of Business & Economic Statistics, vol. 14, no. 4, pages 429–434, 1996.

[Jasra 07] A. Jasra, D.A. Stephens & C.C. Holmes. *On population-based simulation for static inference.* Statistics and Computing, vol. 17, no. 3, pages 263–279, 2007.

[Kim 98] S. Kim, N. Shephard & S. Chib. *Stochastic volatility: likelihood inference and comparison with ARCH models.* Rev. Econ. Studies, vol. 65, no. 3, pages 361–393, 1998.

[Kitagawa 98] G. Kitagawa. *A Self-Organizing State-Space Model.* J. Am. Statist. Assoc., vol. 93, pages 1203–1215, 1998.

[Koop 07] G. Koop & S. M. Potter. *Forecasting and Estimating Multiple Change-point models with an Unknown Number of Change-points.* Review of Economic Studies, vol. 74, pages 763 – 789, 2007.

[Künsch 01] H. Künsch. *State Space and Hidden Markov Models.* In O. E. Barndorff-Nielsen, D. R. Cox & C. Klüppelberg, editeurs, Complex Stochastic Systems, pages 109–173. Chapman and Hall, 2001.

[Liu 98] J. Liu & R. Chen. *Sequential Monte Carlo methods for dynamic systems.* J. Am. Statist. Assoc., vol. 93, pages 1032–1044, 1998.

[Liu 01] J. Liu & M. West. *Combined parameter and state estimation in simulation-based filtering.* In A. Doucet, N. de Freitas & N. J. Gordon, editeurs, Sequential Monte Carlo Methods in Practice, pages 197–223. Springer-Verlag, 2001.

[Liu 08] Jun S. Liu. Monte Carlo strategies in scientific computing. Springer Series in Statistics. Springer, New York, 2008.

[Neal 01] R. M. Neal. *Annealed importance sampling.* Statist. Comput., vol. 11, pages 125–139, 2001.

[Oudjane 05] N. Oudjane & S. Rubenthaler. *Stability and Uniform Particle Approximation of Nonlinear Filters in Case of Non Ergodic Signals.* Stochastic Analysis and applications, vol. 23, pages 421–448, 2005.

[Papaspiliopoulos 07] Omiros Papaspiliopoulos, Gareth O. Roberts & Martin Sköld. *A general framework for the parametrization of hierarchical models.* Statist. Sci., vol. 22, no. 1, pages 59–73, 2007.

[Peters 10] G.W. Peters, G.R. Hosack & K.R. Hayes. *Ecological non-linear state space model selection via adaptive particle Markov chain Monte Carlo.* Arxiv preprint arXiv:1005.2238, 2010.

[Roberts 04] Gareth O. Roberts, Omiros Papaspiliopoulos & Petros Dellaportas. *Bayesian inference for non-Gaussian Ornstein-Uhlenbeck stochastic volatility processes.* J. R. Stat. Soc. Ser. B Stat. Methodol., vol. 66, no. 2, pages 369–393, 2004.

[Robinson 95] M.E. Robinson & J.A. Tawn. *Statistics for exceptional athletics records.* Applied Statistics, vol. 44, no. 4, pages 499–511, 1995.

[Silva 09] R. Silva, P. Giordani, R. Kohn & M. Pitt. *Particle filtering within adaptive Metropolis Hastings sampling.* Arxiv preprint arXiv:0911.0230, 2009.

[Storvik 02] G. Storvik. *Particle filters for state-space models with the presence of unknown static parameters.* IEEE Transaction on Signal Processing, vol. 50, pages 281–289, 2002.

[Whiteley 11] N. Whiteley. *Stability properties of some particle filters.* Arxiv preprint arXiv:1109.6779, 2011.

A Proof of Proposition 1

We remark first that $\psi_{t,\theta}$ may be rewritten as follows:

$$\psi_{t,\theta}(x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}) = \left\{\prod_{n=1}^{N_x} q_{1,\theta}\left(x_1^n\right)\right\}\left\{\prod_{s=2}^{t}\prod_{n=1}^{N_x} W_{s-1,\theta}^{a_{s-1}^n} q_{s,\theta}\left(x_s^n | x_{s-1}^{a_{s-1}^n}\right)\right\}.$$

Starting from (6.7) and (6.3), one obtains

$$\begin{aligned}
\pi_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}) &= \frac{p(\theta)\psi_{t,\theta}(x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x})}{N_x^t p(y_{1:t})} \prod_{s=1}^{t}\left\{\sum_{n=1}^{N_x} w_{s,\theta}(x_{s-1}^{a_{s-1}^n}, x_s^n)\right\} \\
&= \frac{p(\theta)}{N_x^t p(y_{1:t})} \sum_{n=1}^{N_x}\left[w_{t,\theta}(x_{t-1}^{a_{t-1}^n}, x_t^n)\left\{\prod_{i=1}^{N_x} q_{1,\theta}(x_1^i)\right\}\right. \\
&\qquad \left.\left\{\prod_{s=2}^{t}\prod_{i=1}^{N_x} W_{s-1,\theta}^{a_{s-1}^i} q_{s,\theta}(x_s^i | x_{s-1}^{a_{s-1}^i})\right\}\prod_{s=1}^{t-1}\left\{\sum_{i=1}^{N_x} w_{s,\theta}(x_{s-1}^{a_{s-1}^i}, x_s^i)\right\}\right]
\end{aligned}$$

by distributing the final product in the first line, and using the convention that $w_{1,\theta}(x_0^{a_0^n}, x_1^n) = w_{1,\theta}(x_1^n)$.

To obtain (6.8), we consider the summand above, for a given value of n, and put aside the random variables that correspond to the state trajectory $\mathbf{x}_{1:t}^n$. We start with $\mathbf{x}_{1:t}^n(t) = x_t^n$, and note that

$$\begin{aligned}
w_{t,\theta}(x_{t-1}^{a_{t-1}^n}, x_t^n) q_{t,\theta}(x_t^n | x_{t-1}^{a_{t-1}^n}) &= f_\theta(x_t^n | x_{t-1}^{a_{t-1}^n}) g_\theta(y_t | x_t^n) \\
&= f_\theta\left(\mathbf{x}_{1:t}^n(t) | \mathbf{x}_{1:t}^n(t-1)\right) g_\theta\left(y_t | \mathbf{x}_{1:t}^n(t)\right),
\end{aligned}$$

and that

$$W_{t-1,\theta}^{a_{t-1}^n}\left\{\sum_{i=1}^{N_x} w_{t-1,\theta}(x_{t-2}^{a_{t-2}^i}, x_{t-1}^i)\right\} = w_{t-1,\theta}\left(\mathbf{x}_{1:t}^n(t-2), \mathbf{x}_{1:t}^n(t-1)\right).$$

Thus, the summand in the expression of π_t above may be rewritten as

$$w_{t-1,\theta}(\mathbf{x}_{1:t}^n(t-2), \mathbf{x}_{1:t}^n(t-1)) f_\theta\left(\mathbf{x}_{1:t}^n(t) | \mathbf{x}_{1:t}^n(t-1)\right) g_\theta\left(y_t | \mathbf{x}_{1:t}^n(t)\right)\left\{\prod_{i=1}^{N_x} q_{1,\theta}(x_1^i)\right\} \times$$

$$\left\{\prod_{s=2}^{t-1}\prod_{i=1}^{N_x} W_{s-1,\theta}^{a_{s-1}^i} q_{s,\theta}(x_s^i | x_{s-1}^{a_{s-1}^i})\right\}\left\{\prod_{\substack{i=1 \\ i \neq \mathbf{h}_t^n(n)}}^{N_x} W_{t-1,\theta}^{a_{t-1}^i} q_{t,\theta}(x_t^i | x_{t-1}^{a_{t-1}^i})\right\}\prod_{s=2}^{t-1}\left\{\sum_{i=1}^{N_x} w_{s-1,\theta}(x_{s-2}^{a_{s-2}^n}, x_{s-1}^i)\right\}.$$

By applying recursively, for $s = t-1, \ldots, 1$ the same type of substitutions, that is,

$$\begin{aligned}
w_s\left(\mathbf{x}_{1:t}^n(s-1), \mathbf{x}_{1:t}^n(s)\right) q_{s,\theta}(x_s^{\mathbf{h}_t^n(s)} | x_{s-1}^{\mathbf{h}_t^n(s-1)}) &= f_\theta\left(\mathbf{x}_{1:t}^n(s) | \mathbf{x}_{1:t}^n(s-1)\right) g_\theta\left(y_s | \mathbf{x}_{1:t}^n(s)\right), \\
w_1\left(\mathbf{x}_1^n(1)\right) q_{1,\theta}(x_1^{\mathbf{h}_t^n(1)}) &= \mu_\theta\left(\mathbf{x}_{1:t}^n(1)\right) g_\theta\left(y_1 | \mathbf{x}_{1:t}^n(1)\right),
\end{aligned}$$

and, for $s \geq 2$,

$$W_{s-1,\theta}^{a_{s-1}^n}\left\{\sum_{i=1}^{N_x} w_{s-1,\theta}(x_{s-2}^{a_{s-2}^i}, x_{s-1}^i)\right\} = w_{s-1,\theta}\left(\mathbf{x}_{1:t}^n(s-2), \mathbf{x}_{1:t}^n(s-1)\right).$$

and noting that

$$\begin{aligned} p(\theta, \mathbf{x}_{1:t}^n, y_{1:t}) &= p(y_{1:t})p(\theta|y_{1:t})p(\mathbf{x}_{1:t}^n|y_{1:t}, \theta) \\ &= p(\theta) \prod_{s=1}^{t} \left\{ f_\theta\left(\mathbf{x}_{1:t}^n(s)|\mathbf{x}_{1:t}^n(s-1)\right) g_\theta\left(y_s|\mathbf{x}_{1:t}^n(s)\right) \right\}, \end{aligned}$$

where $p(\theta, \mathbf{x}_{1:t}^n, y_{1:t})$ stands for the joint probability density defined by the model, for the triplet of random variables $(\theta, x_{1:t}, y_{1:t})$, evaluated at $x_{1:t} = \mathbf{x}_{1:t}^n$, one eventually gets:

$$\begin{aligned} &\pi_t(\theta, x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}) = p(\theta|y_{1:t})\times \\ &\qquad \frac{1}{N_x^t} \sum_{n=1}^{N_x} p(\mathbf{x}_{1:t}^n|\theta, y_{1:t}) \left\{ \prod_{\substack{i=1 \\ i \neq \mathbf{h}_t^n(1)}}^{N_x} q_{1,\theta}(x_1^i) \right\} \left\{ \prod_{s=2}^{t} \prod_{\substack{i=1 \\ i \neq \mathbf{h}_t^n(s)}}^{N_x} W_{s-1,\theta}^{a_{s-1}^i} q_{s,\theta}(x_s^i|x_{s-1}^{a_{s-1}^i}) \right\}. \end{aligned}$$

B Proof of Proposition 2 and discussion of assumptions

Since $p(\theta|y_{1:t})$ is the marginal distribution of $\bar{\pi}_{t,t+p}$, by iterated conditional expectation we get:

$$\begin{aligned} \frac{p(y_{t+1:t+p}|y_{1:t})^2}{\mathcal{E}_{t,t+p}^{N_x}} &= \mathbb{E}_{p(\theta|y_{1:t})}\left[\mathbb{E}_{\bar{\pi}_{t,t+p}(\cdot|\theta)}\left\{\hat{Z}_{t+p|t}^2(\theta, x_{1:t+p}^{1:N_x}, a_{1:t+p-1}^{1:N_x})\right\}\right] \\ &= \mathbb{E}_{p(\theta|y_{1:t})}\left[\mathbb{E}_{\pi_t(\cdot|\theta)}\left\{\mathbb{E}_{\psi_{t+p|t}(\cdot|x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}\theta)}\left\{\hat{Z}_{t+p|t}^2(\theta, x_{1:t+p}^{1:N_x}, a_{1:t+p-1}^{1:N_x})\right\}\right\}\right]. \end{aligned}$$

To study the inner expectation, we make the following first set of assumptions:

(H1a) For all $\theta \in \Theta$, and $x, x', x'' \in \mathcal{X}$,

$$\frac{f_\theta(x|x')}{f_\theta(x|x'')} \leq \beta.$$

(H1b) For all $\theta \in \Theta$, $x, x' \in \mathcal{X}$, $y \in \mathcal{Y}$,

$$\frac{g_\theta(y|x)}{g_\theta(y|x')} \leq \delta.$$

Under these assumptions, one obtains the following non-asymptotic bound.

Proposition 3 (Theorem 1.5 of [Cérou 11]). *For $N_x \geq \beta\delta p$,*

$$\mathbb{E}_{\psi_{t+p|t}(\cdot|x_{1:t}^{1:N_x}, a_{1:t-1}^{1:N_x}\theta)}\left\{\frac{\hat{Z}_{t+p|t}^2(\theta, x_{1:t+p}^{1:N_x}, a_{1:t+p-1}^{1:N_x})}{p(y_{t+1:t+p}|y_{1:t}, \theta)^2}\right\} - 1 \leq 4\beta\delta\frac{p}{N_x}.$$

The Proposition above is taken from [Cérou 11], up to some change of notations and a minor modification: [Cérou 11] establish this result for the likelihood estimate $\hat{Z}_t$, obtained by running a particle from time 1 to time t. However, their proof applies straightforwardly to the partial likelihood estimate $\hat{Z}_{t+p|t}$, obtained by running a particle filter from time $t+1$ to time $t+p$, and therefore with initial distribution η_0 set to the mixture of Dirac masses at the particle locations at time t. We note in passing that Assumptions (H1a) and (H1b) may be loosened up slightly, see [Whiteley 11]. A direct consequence of Proposition 3, the main definitions and the iterated expectation is Proposition 2(a) for $\eta = 4\beta\delta$.

Proposition 2(b) requires a second set of conditions taken from [Chopin 02]. These relate to the asymptotic behaviour of the marginal posterior distribution $p(\theta|y_{1:t})$ and they have been used to study the weight degeneracy of IBIS. Let $l_t(\theta) = \log p(y_{1:t}|\theta)$. The following assumptions hold almost surely.

(H2a) The MLE $\hat{\theta}_t$ (the mode of function $l_t(\theta)$) exists and converges to θ_0 as $n \to +\infty$.

(H2b) The observed information matrix defined as

$$\Sigma_t = -\frac{1}{t}\frac{\partial l_t(\hat{\theta}_t)}{\partial\theta\partial\theta'}$$

is positive definite and converges to $I(\theta_0)$, the Fisher information matrix.

(H2c) There exists Δ such that, for $\delta \in (0, \Delta)$,

$$\limsup_{t\to+\infty}\left[\frac{1}{t}\sup_{\|\theta-\hat{\theta}_t\|>\delta}\left\{l_t(\theta) - l_t(\hat{\theta}_t)\right\}\right] < 0.$$

(H2d) The function l_t/t is six-times continuously differentiable, and its derivatives of order six are bounded relative to t over any compact set $\Theta' \subset \Theta$.

Under these conditions one may apply Theorem 1 of [Chopin 02] [see also Proof of Theorem 4 in Chopin 04] to conclude that $\mathcal{E}^{\infty}_{t,t+p} \geq 2\gamma$ for a given $\gamma > 0$ and t large enough, provided $p = \lceil \tau t \rceil$ for some $\tau > 0$ (that depends on γ). Together with Proposition 2(a) and by a small modification of the Proof of Theorem 4 in [Chopin 04] to fix γ instead of τ, we obtain Proposition 2(b) provided $N_x = \lceil \eta t \rceil$, and $\eta = 4\beta\delta$.

Note that (H2a) and (H2b) essentially amount to establishing that the MLE has a standard asymptotic behaviour (such as in the IID case). This type of results for state-space models is far from trivial, owning among other things to the intractable nature of the likelihood $p(y_{1:t}|\theta)$. A good entry in this field is Chapter 12 of [Cappé 05], where it can be seen that the first set of conditions above, (H1a) and (H2b), are sufficient conditions for establishing (H2a) and (H2b), see in particular Theorem 12.5.7 page 465. Condition (H2d) is trivial to establish, if one assumes bounds similar to those in (H1a) and (H1b) for the derivatives of g_θ and f_θ. Condition (H2c) is harder to establish. We managed to prove that this condition holds for a very simple linear Gaussian model; notes are available from the first author. Independent work by Judith Rousseau and Elisabeth Gassiat is currently carried out on the asymptotic properties of posterior distributions of state-space models, where (H2c) is established under general conditions (personal communication).

The implication of Proposition 2 to the stability of SMC^2 is based on the additional assumption that after resampling at time t we obtain exact samples from π_t. In practice, this is only approximately true since an MCMC scheme is used to sample new particles. This assumption also underlies the analysis of IBIS in [Chopin 02], where it was demonstrated empirically (see e.g. Fig. 1(a) in that paper) that the MCMC kernel which updates the θ particles has a stable efficiency over time since it uses the population of θ-particles to design the proposal distribution. We also observe empirically that the performance of the PMCMC step does not deteriorate over time provided N_x is increased appropriately, see for example Figure 6.1(b). It is important to establish such a result theoretically, i.e., that the total variation distance of the PMCMC kernel from the target distribution remains bounded over time provided N_x is increased appropriately. Note that a fundamental difference between IBIS and SMC^2 is that respect is that in the latter the MCMC step targets distributions in increasing dimensions as time increases. Obtaining such a theoretical result is a research project on its own right, since such quantitative results lack, to the best of our knowledge, from the existing literature. The closest in spirit is Theorem 6 in [Andrieu 09] which, however, holds for "large enough" N_x, instead of providing a quantification of how large N_x needs to be.

Chapter 7

Future lines of research

Ce dernier chapitre propose des pistes de recherche futures, dans la continuité des travaux présentés dans les chapitres précédents.

D'une part, les travaux concernant l'algorithme de Wang–Landau peuvent être continués dans les directions suivantes. Les résultats de convergence obtenus dans le chapitre 4 pourraient être étendus afin d'obtenir la convergence du processus joint, constitué de la chaîne $(X_t)_{t\geq 0}$ et du processus de pénalité $(\theta_t)_{t\geq 0}$. L'étude de ce processus joint, qui constitue une chaîne de Markov, permettrait non seulement de retrouver les résultats obtenus sous des hypothèses nettement plus réalistes, mais aussi d'aboutir à une version de l'algorithme plus élégante, qui ne nécessiterait pas de choisir une séquence de tailles de pas. Un tel algorithme serait plus simple à paramétrer et éventuellement plus performant que la méthode originale. Par ailleurs, la variante de l'algorithme utilisant plusieurs chaînes en parallèle, présentée dans le chapitre 5, exige un cadre théorique différent. En effet, elle peut être vue comme l'approximation en champ moyen d'un algorithme idéal qui utiliserait une infinité de chaînes. Cette approche permettrait de quantifier l'effet du nombre de chaînes, et ainsi de pouvoir donner des recommendations sur le nombre de chaînes optimal à utiliser, à coût computationnel donné.

D'autre part, les bonnes performances de l'algorithme présenté dans le chapitre 6 motivent une étude plus théorique. Notamment, l'analyse faite de sa complexité algorithmique dans le chapitre précédent repose sur des hypothèses fortes, qui sont rarement vérifiées en pratique. Dans la continuité de travaux récents dans la littérature sur les méthodes de Monte Carlo séquentiel, il serait intéressant d'étudier le coût de la méthode sous des hypothèses réalistes et facilement vérifiables. Par ailleurs, l'estimateur de l'évidence fourni par l'algorithme est similaire à l'estimateur de la vraisemblance fourni par les méthodes particulaires dans le contexte du filtrage. Une étude théorique de la variance de cet estimateur en fonction du nombre d'observations viendrait justifier théoriquement l'utilisation de la méthode SMC^2 pour le calcul des facteurs de Bayes dans les modèles à espace d'états.

The lines of research presented in this document can be pursued in various directions, which are presented in this chapter.

7.1 On the Wang–Landau algorithm

The Wang–Landau algorithm in its original form is validated by the theory on stochastic approximation algorithms, see [Andrieu 05] and also [Liang 11]. However some aspects are still to be studied in details, notably the behaviour of the algorithm when the diminishing step-size is replaced by a fixed value, and the stability of the algorithm as the number of parallel chains increases.

7.1.1 Removing the diminishing step-size

Recall that the Wang–Landau algorithm (with diminishing step-sizes) generates two objects:

- a sample $(X_t)_{t\geq 0}$ such that the bins are visited according to desired frequencies $(\phi^{(i)})_{i=1}^{d}$ specified by the user, and the empirical distribution of the sample within each bin converges towards the restriction of the target distribution π to this bin,
- and a process of penalties $(\theta_t)_{t\geq 0} = (\theta_t^{(1)}, \ldots, \theta_t^{(d)})_{t\geq 0}$ converging to some d-dimensional vector $\theta^\star$ when t goes to infinity.

In order to prove that the Flat Histogram criterion is met in finite time, Chapter 4 considers the behaviour of the Wang–Landau algorithm when the sequence of step-sizes, $(\gamma_t)_{t\geq 0}$, is set to a fixed value $\gamma > 0$. In this setting, it proposes a proof that the generated sample $(X_t)_{t\geq 0}$ visits each bin $\mathcal{X}^{(i)}$ at the desired frequency $\phi^{(i)}$, for any bin index $i \in \{1, \ldots, d\}$. Formally:

$$\frac{1}{t}\sum_{n=1}^{t} \mathbb{1}_{\mathcal{X}^{(i)}}(X_n) \xrightarrow[t\to\infty]{\mathbb{L}_1} \phi^{(i)}$$

Hence, for a fixed $\gamma > 0$ (i.e. without *diminishing adaptation*), the generated sample already verifies some of the desired properties listed above. However, from Chapter 4 nothing can be directly said about the limiting distribution (if it exists) of the sample $(X_t)_{t\geq 0}$ when the step-size γ is fixed. Likewise, the process of penalties $(\theta_t)_{t\geq 0}$ does not converge to a fixed value $\theta^\star$, but instead might converge or not to a distribution.

This leads to the following question: is it really useful to decrease the sequence of step-sizes $(\gamma_t)_{t\geq 0}$? Considering that the joint process $(X_t, \theta_t)_{t\geq 0}$ is a Markov chain, does it converge to an unique invariant distribution when γ is fixed? If so, then what is the marginal distribution of the X-component? Is the distribution of the θ-component related to the limiting value $\theta^\star$ of the original algorithm, for example is $\theta^\star$ equal to the expectation of that distribution? A thorough study of these questions might allow to remove the sequence of step-sizes from the tuning parameters. Since it is known to be a sensitive parameter in practice, removing it might lead to a more efficient algorithm.

7.1.2 Number of chains

Chapter 5 presents an extension of the Wang–Landau algorithm where N chains explore the state space $\mathcal{X}$ in parallel, sharing the same penalty process $(\theta_t)_{t\geq 0}$. Numerical examples show that the number of chains has a impact on the stability of the penalty process (see e.g. Figure 5.3).

To quantify the effect of the number of chains, one could use techniques borrowed from the particle filter literature, describing the propagation of errors through Feynman–Kac formulæ [Del Moral 04]. This study would require at least the two following steps:

- to identify the limiting system that the algorithm with N chains is mimicking, and to analyse the asymptotic behaviour of the limiting system when time goes to infinity,
- to control the error between the N-chains approximation and the limiting system.

Towards the first step, we can show using basic manipulations that the limiting system has a stable asymptotic behaviour. For instance consider the following update of the penalties, introduced in equation (4.2) and generalised to N chains:

$$\log \theta_t^{(i)} = \log \theta_{t-1}^{(i)} + \log \left[1 + \gamma \left(\frac{1}{N} \sum_{k=1}^{N} \mathbb{I}_{\mathcal{X}^{(i)}}(X_t^{(k)}) - \phi^{(i)} \right) \right]$$

In the case where $\phi^{(i)} = 1/d$ for all bin index i, consider the normalised penalties:

$$\tilde{\theta}_t^{(i)} = \frac{\theta_t^{(i)}}{\sum_{j=1}^{d} \theta_t^{(j)}}$$

If N goes to infinity and if the chains are sampled iid from the target distribution π_{θ_t} at time t, one can show that the normalised penalties satisfy the simple recursion:

$$\tilde{\theta}_t^{(i)} = \tilde{\theta}_{t-1}^{(i)} \times (1 - \alpha_{t-1}) + \alpha_{t-1} \times \psi^{(i)}$$

with

$$\psi^{(i)} = \int_{\mathcal{X}^{(i)}} \pi(x) dx$$

and some sequence $(\alpha_t)_{t \geq 0}$ in $[0, 1]$. Then, assuming that $\theta^{(i)} > 0$, $\psi^{(i)} > 0$ for all bin index i, one can bound α_t away from 0, and obtain the following geometric convergence of $\tilde{\theta}_t$ towards ψ:

$$\exists \rho < 1 \; \exists C > 0 \; \forall i \in \{1, \ldots, d\} \; \forall t \geq 0 \quad \|\tilde{\theta}_t - \psi\| < C\rho^t$$

Numerical experiments show that the penalty process of the algorithm that uses N chains behaves very much like its ideal counterpart, when N is large enough. Indeed the penalty process seems to converge towards a neighborhood of ψ, which radius diminishes with N, instead of converging directly to ψ like in the ideal version of the algorithm. Therefore we can hope that the literature on Feynman–Kac formulæ and mean-field approximations provides methods to analyse the difference between the N-chains approximation and the limiting system corresponding an infinite number of chains.

As a result, one might grasp some understanding about the impact of the number of chains N on the stabilisation of the penalty process, which could lead to guidelines on how to choose N, for a fixed computational cost: for a given number of target density evaluations $C = N \times T$, how to choose the number of chains N and the number of iterations T?

7.2 On Bayesian inference in Hidden Markov models

Chapter 6 introduces a sequential method to sample from the posterior distribution of Hidden Markov models, requiring only that one can sample from the transition distribution and evaluate the measurement probability density function up to a multiplicative constant. This setting encompasses many models, if not all: for instance it precludes cases where the measurement distribution is not tractable. In particular it contains the numerous cases where the hidden process comes from the discretisation of a stochastic differential equation, and the observations are noisy measurements of the hidden process where the noise is assumed to follow a simple parametric distribution.

While sequential Monte Carlo methods are applicable in many cases, in the sense that they can be implemented on a computer, their theoretical justification usually relies on assumptions that are not met in practice. There is therefore a gap between theoretical results on the topic and practical cases. Recently, some effort has been pursued to relax these usual assumptions, see e.g. [Whiteley 11] and references therein.

7.2.1 Validation of SMC^2 under verifiable assumptions

In this context, SMC^2 could be studied in the light of general assumptions. The justification of the method in Chapter 6 covers its consistency: the method is targeting the correct sequence of distributions, and hence would provide the exact quantities of interest when the number of particles goes to infinity. This part does not rely on specific assumptions on the model. The justification also includes a study of the computational cost, which relies on two types of results:

- results bounding the non-asymptotic variance of the normalising constant in particle filters, which allow to scale the number of particles with the number of observations,
- and classical results on sequential Monte Carlo samplers, showing the stability of the method when the posterior distribution concentrates around the Maximum Likelihood Estimate.

In its current form, the validation of SMC^2 uses results taken from [Cérou 11] for the variance of the normalising constant estimate, and from [Chopin 02] for the stability of SMC samplers when the posterior distribution concentrates. Both of these results rely on assumptions that are rarely met in general HMMs, and in particular these assumptions are not verified for the very examples of Chapter 6. Mild assumptions such as the ones used in [Whiteley 11] or [Douc 09] might be sufficient to justify why the SMC^2 algorithm works in practice and how the computational cost scales with the number of observations. In particular, [Douc 09] do not rely on the assumption that the observations were generated from the model, which is particularly important in practice when dealing with real data; it would therefore be interesting to establish the validity of SMC^2 under a similar setting.

7.2.2 Estimation of the model evidence

A related problem is the theoretical study of the estimate of the model evidence provided by SMC^2. Recall from section 6.3.5 that the method provides an estimate of the evidence at each time step, as illustrated on Figure 6.3(c), similarly to particle filters that provide estimates of the likelihood given a parameter value, as reminded in Equation (6.3) These estimates of the likelihood, which are called normalising constants in the Feynman–Kac literature, are precisely the objects studied e.g. in [Cérou 11].

This leads to obvious questions: what can we say about the variance of the evidence estimate? Can we bound it by a function of the number of observations t, the number of x-particles N_x and the number of θ-particles N_θ? What would be the equivalent of the results of [Cérou 11] in this context?

Those questions were not addressed in Chapter 6 but might have some practical importance if SMC^2 proves to be an efficient method for estimating the evidence, which are building blocks for model choice under the Bayesian paradigm since they appear in the computation of Bayes factors. Since particle filters are considered to be efficient methods to estimate the likelihood for a given parameter value, there is some hope that SMC^2 would be a serious competitor when estimating the evidence of a HMM.

The estimation of the evidence could therefore be worth an empirical study comparing SMC^2 with other methods based on the output of particle Markov chain Monte Carlo methods, as well as a more theoretical study on the variance of this estimate.

Bibliography

[Andrieu 05] C. Andrieu, E. Moulines & P. Priouret. *Stability of stochastic approximation under verifiable conditions.* SIAM Journal on control and optimization, vol. 44, no. 1, pages 283–312, 2005.

[Cérou 11] F. Cérou, P. Del Moral & A. Guyader. *A nonasymptotic theorem for unnormalized Feynman–Kac particle models.* Ann. Inst. Henri Poincarré, vol. 47, no. 3, pages 629–649, 2011.

[Chopin 02] N. Chopin. *A sequential particle filter method for static models.* Biometrika, vol. 89, pages 539–552, 2002.

[Del Moral 04] P Del Moral. Feynman-kac formulae. Springer, 2004.

[Douc 09] R. Douc, G. Fort, E. Moulines & P. Priouret. *Forgetting the initial distribution for Hidden Markov Models.* Stochastic Processes and their Applications, vol. 119, no. 4, pages 1235–1256, 2009.

[Liang 11] F. Liang & M. Wu. Population stochastic approximation mcmc algorithm and its weak convergence. 2011.

[Whiteley 11] N. Whiteley. *Stability properties of some particle filters.* ArXiv e-prints, September 2011.