Efficient Bayesian inference via Monte Carlo and machine learning algorithms

by

Fernando Llorente Fernández

A dissertation submitted by in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

Mathematical Engineering

Universidad Carlos III de Madrid

Advisors:

David Delgado Gómez
Luca Martino

November, 2022

ACKNOWLEDGEMENTS

I would like to sincerely thank my supervisors David Delgado Gómez and Luca Martino. This thesis would have not been possible without them. During the last four years, I have learned immensely from David and Luca. I have learned a lot of technical knowledge. I have learned how to do research. I have learned about life in the Academia. More importantly, I have learned about life in general. I will be always grateful for their invaluable advise and support, and their time! I also want to thank my coauthors and colleagues with whom I shared many discussions and ideas. I deeply thank my parents and friends for always being there. Finally, I thank all the people, present and past, that in some way or another have supported me during this journey.

ABSTRACT

In many fields of science and engineering, we are faced with an inverse problem where we aim to recover an unobserved parameter or variable of interest from a set of observed variables. Bayesian inference is a probabilistic approach for inferring this unknown parameter that has become extremely popular, finding application in myriad problems in fields such as machine learning, signal processing, remote sensing and astronomy. In Bayesian inference, all the information about the parameter is summarized by the posterior distribution. Unfortunately, the study of the posterior distribution requires the computation of complicated integrals, that are analytically intractable and need to be approximated. Monte Carlo is a huge family of sampling algorithms for performing optimization and numerical integration that has become the main horsepower for carrying out Bayesian inference. The main idea of Monte Carlo is that we can approximate the posterior distribution by a set of samples, obtained by an iterative process that involves sampling from a known distribution. Markov chain Monte Carlo (MCMC) and importance sampling (IS) are two important groups of Monte Carlo algorithms. This thesis focuses on developing and analyzing Monte Carlo algorithms (either MCMC, IS or combination of both) under different challenging scenarios presented below. In summary, in this thesis we address several important points, enumerated **(a)**–**(f)**, that currently represent a challenge in Bayesian inference via Monte Carlo. A first challenge that we address is the problematic exploration of the parameter space by off-the-shelf MCMC algorithms when there is **(a)** multimodality, or with **(b)** highly concentrated posteriors. Another challenge that we address is the **(c)** proposal construction in IS. Furtheremore, in recent applications we need to deal with **(d)** expensive posteriors, and/or we need to handle **(e)** noisy posteriors. Finally, the Bayesian framework also offers a way of comparing competing hypothesis (models) in a principled way by means of marginal likelihoods. Hence, a task that arises as of fundamental importance is **(f)** marginal likelihood computation.

Chapters 2 and 3 deal with **(a)**, **(b)**, and **(c)**. In Chapter 2, we propose a novel population MCMC algorithm called Parallel Metropolis-Hastings Coupler (PMHC). PMHC is very suitable for multimodal scenarios since it works with a population of states, instead of a single one, hence allowing for sharing information. PMHC combines independent exploration by the use of parallel Metropolis-Hastings algorithms, with cooperative exploration by the use of a population MCMC technique called Normal Kernel Coupler. In Chapter 3, population MCMC are combined with IS within the layered adaptive IS (LAIS) framework. The combination of MCMC and IS serves two purposes. First, an automatic proposal construction. Second, it aims at increasing the robustness, since the MCMC samples are not used directly to form the sample approximation of the posterior. The use of minibatches of data is proposed to deal with highly concentrated posteriors. Other extensions for reducing the costs with respect to the vanilla LAIS framework, based

on recycling and clustering, are discussed and analyzed.

Chapters 4, 5 and 6 deal with **(c)**, **(d)** and **(e)**. The use of nonparametric approximations of the posterior plays an important role in the design of efficient Monte Carlo algorithms. Nonparametric approximations of the posterior can be obtained using machine learning algorithms for nonparametric regression, such as Gaussian Processes and Nearest Neighbors. Then, they can serve as cheap surrogate models, or for building efficient proposal distributions. In Chapter 4, in the context of expensive posteriors, we propose adaptive quadratures of posterior expectations and the marginal likelihood using a sequential algorithm that builds and refines a nonparametric approximation of the posterior. In Chapter 5, we propose Regression-based Adaptive Deep Importance Sampling (RADIS), an adaptive IS algorithm that uses a nonparametric approximation of the posterior as the proposal distribution. We illustrate the proposed algorithms in applications of astronomy and remote sensing. Chapter 4 and 5 consider noiseless posterior evaluations for building the nonparametric approximations. More generally, in Chapter 6 we give an overview and classification of MCMC and IS schemes using surrogates built with noisy evaluations. The motivation here is the study of posteriors that are both costly and noisy. The classification reveals a connection between algorithms that use the posterior approximation as a cheap surrogate, and algorithms that use it for building an efficient proposal. We illustrate specific instances of the classified schemes in an application of reinforcement learning. Finally, in Chapter 7 we study noisy IS, namely, IS when the posterior evaluations are noisy, and derive optimal proposal distributions for the different estimators in this setting.

Chapter 8 deals with **(f)**. In Chapter 8, we provide with an exhaustive review of methods for marginal likelihood computation, with special focus on the ones based on Monte Carlo. We derive many connections among the methods and compare them in several simulations setups. Finally, in Chapter 9 we summarize the contributions of this thesis and discuss some potential avenues of future research.

PUBLISHED AND SUBMITTED CONTENT

The materials from the following sources are included in the thesis. Their inclusion is not indicated by typographical means or references, since they are fully embedded in each chapter indicated below.

Paper A. F. Llorente, L. Martino, D. Delgado (2019). "Parallel-Metropolis Hastings Coupler". *IEEE Signal Processing Letters*, 26, 953–957.

- DOI: 10.1109/LSP.2019.2913470
- Included in Chapter 2 (Full)

Paper B. F. Llorente, E. Curbelo, L. Martino, V. Elvira, D. Delgado (2022). "MCMC-driven importance samplers". *Applied Mathematical Modelling*, 111, 310–331

- DOI: 10.1016/j.apm.2022.06.027
- Included in Chapter 3 (Full)

Paper C. F. Llorente, L. Martino, V. Elvira, D. Delgado, J. López-Santiago (2020). "Adaptive quadrature schemes for Bayesian inference via active learning". *IEEE Access*, 8, 208462-208483.

- DOI: 10.1109/ACCESS.2020.3038333
- Included in Chapter 4 (Full)

Paper D. F. Llorente, L. Martino, D. Delgado-Gómez, G. Camps-Valls . "Deep Importance Sampling based on Regression for Model Inversion and Emulation". *Digital Signal Processing*, 116, 103104.

- DOI: 10.1016/j.dsp.2021.1031043
- Included in Chapter 5 (Full)

Paper E. F. Llorente, L. Martino, J. Read, D. Delgado (2021). "A survey of Monte Carlo methods for noisy and costly densities with application to reinforcement learning". *arXiv preprint*, arXiv:2108.00490

- URL: 2108.00490 (submitted for publication)
- Included in Chapter 6 (Full)

Paper F. F. Llorente, L. Martino, J. Read, D. Delgado (2022). "Optimality in Noisy Importance Sampling". *Signal Processing*, 194, 108455

- DOI: 10.1016/j.sigpro.2022.108455

- Included in Chapter 7 (Full)

Paper G. F. Llorente, L. Martino, D. Delgado (2023). "Marginal likelihood computation for model selection and hypothesis testing: an extensive review". *SIAM Review (to appear)*.

- URL: 2005.08334
- Included in Chapter 8 (Full)

FURTHER RESEARCH ACHIEVEMENTS

Papers:

1. F. Llorente, L. Martino, E. Curbelo, J. Lopez-Santiago, D. Delgado (2022). "On the safe use of prior densities for Bayesian model selection". *Wiley Interdisciplinary Reviews: Computational Statistics*.
   DOI: 10.1002/wics.1595.

2. F. Llorente, L. Martino, D. Delgado-Gómez (2021). Contributed discussion to "On a Class of Objective Priors from Scoring Rules". *Bayesian Analysis*.
   DOI: 10.1214/19-BA1187

3. L. Martino, F. Llorente, E. Curbelo, J. Lopez-Santiago, J. Miguez (2021). Automatic tempered posterior distributions for Bayesian inversion problems. *Mathematics*.
   DOI: 10.3390/math9070784

4. R. San Millan-Castino, L. Martino, E. Morgado, F. Llorente (2022). Models of Soundscape Emotions: Rankings and Gibbs Analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.
   DOI: 10.1109/TASLP.2022.3192664

5. D. Carreras-Garcia, D. Delgado-Gómez, F. Llorente, A. Arribas-Gil (2020). Patient no-show prediction: A systematic literature review. *Entropy*.
   DOI: 10.3390/e22060675

Conference papers:

1. F. Llorente, L. Martino, D. Delgado-Gómez, J. Lopez-Santiago (2020). On the computation of marginal likelihood via MCMC for model selection and hypothesis testing. *28th European Signal Processing Conference (EUSIPCO).*
   DOI: 10.23919/Eusipco47968.2020.9287757

2. F. Llorente, L. Martino, V. Elvira, D. Delgado-Gómez (2021). A Nearest Neighbors Quadrature for Posterior Approximation via Adaptive Sequential Design. *IEEE Statistical Signal Processing Workshop (SSP).*
   DOI: 10.1109/SSP49050.2021.9513802

3. F. Llorente, E. Curbelo, L. Martino, P. Olmos, D. Delgado-Gómez (2022). Safe importance sampling based of partial posteriors and neural variational approximations. *30th European Signal Processing Conference (EUSIPCO).*
   URL: pdf

4. M. Chehboune, F. Llorente, R. Kaddah, L. Martino, J. Read (2022). CAMEO: Curiosity Augmented Metropolis for Exploratory Optimal Policies. *30th European Signal Processing Conference (EUSIPCO).*
URL: arxiv

CONTENTS

1. INTRODUCTION

This chapter introduces the preliminary concepts of the thesis. This chapter is organized as follows. Section 1.1 gives a summary of Bayesian inference and introduces the notation used in the chapter. Then, Section 1.2 introduces Monte Carlo methods for the application of Bayesian inference, with special emphasis on the Markov chain Monte Carlo (MCMC) and importance sampling (IS) families. The MCMC and IS when working with noisy realizations of the posterior are briefly discussed in 1.2.4. Section 1.3 address the construction of a nonparametric approximation to the posterior and discuss the potential uses within Monte Carlo. Finally, Section 1.4 describes the structure of the thesis and the main contributions.

1.1. Bayesian inference

This section introduces the basics of the Bayesian inference. In many applications, after receiving a set of observations, the goal is to infer the underlying mechanism that have generated the data. Frequently, a parametric approach is considered. Let $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$ be the parameter of interest and $\mathbf{y} \in \mathbb{R}^{d_y}$ denote the vector of observations. A Bayesian statistical model comprises a likelihood function, $\ell(\mathbf{y}|\mathbf{x}) : \mathbb{R}^{d_y} \times \mathbb{R}^{d_x} \to \mathbb{R}_{\geq 0}$, and a prior probability density function (pdf), $g(\mathbf{x}) : \mathbb{R}^{d_x} \to \mathbb{R}_{\geq 0}$. The conditional distribution of $\mathbf{x}$ given $\mathbf{y}$, namely, the posterior pdf, is given by

$$p(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{\int_{\mathcal{X}} \ell(\mathbf{y}|\mathbf{x}')g(\mathbf{x}')d\mathbf{x}'}. \tag{1.1}$$

Inferences about $\mathbf{x}$ are then obtained in terms of posterior expectations, i.e., by computing integrals of some function $f(\mathbf{x}) : \mathbb{R}^{d_x} \to \mathbb{R}$ with respect to $p(\mathbf{x}|\mathbf{y})$,

$$\mathbb{E}_{p(\mathbf{x}|\mathbf{y})}\left[f(\mathbf{x})\right] = \int_{\mathcal{X}} f(\mathbf{x})p(\mathbf{x}|\mathbf{y})d\mathbf{x}. \tag{1.2}$$

The setting described above is referred to as *parameter estimation* or *level 1 of inference*. The *level 2 of inference* addresses the problem of *model selection* [28, Ch. 28]. Let $\mathcal{M}_i$, $i = 1, \ldots, L$, denote L different models, each one comprising its likelihood function and prior pdf, and with parameters of possibly different dimension. A key quantity of model $\mathcal{M}_i$ is the so-called *marginal likelihood* or *Bayesian evidence*,

$$p(\mathbf{y}|\mathcal{M}_i) = \int_{\mathcal{X}_i} \ell(\mathbf{y}|\mathbf{x}_i, \mathcal{M}_i)g(\mathbf{x}_i|\mathcal{M}_i)d\mathbf{x}_i \,, \tag{1.3}$$

which expresses the probability of the data under model $\mathcal{M}_i$. Note that $p(\mathbf{y}|\mathcal{M}_i)$ is essentially the normalizing constant of the parameter posterior $p(\mathbf{x}_i|\mathbf{y}, \mathcal{M}_i)$, as per Eq. (1.1). The interest lies then in the model posterior,

$$p(\mathcal{M}_i|\mathbf{y}) = \frac{p(\mathbf{y}|\mathcal{M}_i)p(\mathcal{M}_i)}{\sum_{j=1}^{L} p(\mathbf{y}|\mathcal{M}_j)p(\mathcal{M}_j)} \,, \tag{1.4}$$

where $p(\mathcal{M}_j),\ j = 1, \ldots, L$, are prior probabilities associated to the models.

1.1.1. Approximate Bayesian inference

In a nutshell, Bayesian inference revolves around the study of the two posterior distributions in Eqs. (1.1) and (1.4). The applicability hence resides in being able to compute the integrals in Eqs. (1.2) and (1.3). Except for the most trivial settings, these integrals are intractable, hence numerical methods are required.

There are two popular strategies for carrying *approximate* Bayesian inference based on sampling and optimization, each one having its own advantages and disadvantages [42][7, Ch. 10]. The sampling-based approach use stochastic approximations via Monte Carlo, while the optimization-based approach relies on variational inference for approximating probability densities. The focus of this thesis is on Monte Carlo, that is introduced in the next section.

1.1.2. Notation

To simplify notation, from now on the vector of observations $\mathbf{y}$ is assumed to be fixed. Moreover, the focus is restricted to a single model. The following notation will be used in the subsequent sections. The posterior pdf is denoted as $\bar{\pi}(\mathbf{x}) = p(\mathbf{x}|\mathbf{y})$, $\pi(\mathbf{x}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})$ denotes the unnormalized posterior, and $Z = \int \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})d\mathbf{x}$ is the marginal likelihood, such that $\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$.

The notation $\mathbf{x}$ for the parameter of interest will be used throughout this chapter. Note however that, while this is a standard notation in the engineering literature, using the Greek letter $\boldsymbol{\theta}$ to denote model parameters is a more common choice in the statistics literature. This is the reason we use $\boldsymbol{\theta}$, instead of $\mathbf{x}$, in several of the works appearing in this dissertation.

1.2. Monte Carlo methods

This section overviews Monte Carlo, starting from the basic identity and then introducing two families of methods for implementing Monte Carlo in practice, namely, Markov chain Monte Carlo and importance sampling, that are the main focus of this work. Let us formulate the problem of parameter estimation as that of computing the following integral,

$$I = \int_{\mathcal{X}} f(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x}, \tag{1.5}$$

where $f(\mathbf{x})$ is some function of interest (scalar for simplicity). Let $\mathbf{x}_i$ for $i = 1, \ldots, N$, denote a set of independent and identically distributed samples from $\bar{\pi}(\mathbf{x})$. Equivalently,

we write $\{\mathbf{x}_i\}_{i=1}^N \sim \bar{\pi}(\mathbf{x})$. Hence, the above integral can be approximated by the following Monte Carlo estimator,

$$I \approx \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i), \quad \text{where} \quad \{\mathbf{x}_i\}_{i=1}^N \sim \bar{\pi}(\mathbf{x}). \tag{1.6}$$

The Monte Carlo estimator is a random variable with desirable properties such as unbiasedness, asymptotic normality and variance decaying at rate $O(\frac{1}{N})$ [42].

Sample independently from the Bayesian posterior $\bar{\pi}(\mathbf{x})$ is not possible in practice (see [33] for a review of independent random sampling methods). Hence, in the following sections we review Monte Carlo methods that deal with this issue.

1.2.1. Markov chain Monte Carlo

When it is not possible to sample independently from a probability distribution, a popular strategy is instead to generate dependent samples . Markov chain Monte Carlo (MCMC) is a family of Monte Carlo methods that rely on Markov chains for generating dependent samples from a probability distribution [42]. As an illustration, let us consider the well-known Metropolis-Hastings (MH) algorithm. Let $\mathbf{x}_t$ be the current state of chain. A proposed state is sampled from a *proposal* distribution conditional on $\mathbf{x}_t$, $\mathbf{x}' \sim q(\mathbf{x}|\mathbf{x}_t)$. The state $\mathbf{x}'$ is accepted with probability

$$\alpha(\mathbf{x}_t, \mathbf{x}') = \min\left\{1, \frac{\pi(\mathbf{x}')q(\mathbf{x}_t|\mathbf{x}')}{\pi(\mathbf{x}_t)q(\mathbf{x}'|\mathbf{x}_t)}\right\}. \tag{1.7}$$

Note that Z need not be known. The next state of the chain is then $\mathbf{x}_{t+1} = \mathbf{x}'$ when accepted, or, in case of rejection, the current state is repeated, $\mathbf{x}_{t+1} = \mathbf{x}_t$. An analogous of the Monte Carlo estimator in Eq. (1.6) can be then formed by using the resulting set of dependent samples. The efficiency of this MCMC estimator is reduced due to the presence of correlation in the samples. Moreover, although the convergence of this algorithm is ensured under very mild conditions, this convergence can be very slow in practice, such as when the posterior is multimodal or is concentrated is a small region [9, 42].

The literature devoted to improve MCMC algorithms is very vast (see e.g. [27, 43] and references therein). Adaptive MCMC algorithms aim at changing the proposal on the fly based on the history of chain [24, 4]. In this group, several works attempt to build an independent proposal using non-parametric approximations to the posterior [30]. Other, more sophisticated, proposal generating mechanism and transition kernels can be used to build more efficient Markov chains, such as Hamiltonian Monte Carlo and Multiple-try MCMC, among others [9, Ch. 5][22][29]. Tempering and data-tempering are used to artificially increase the variance of the posterior and facilitate the subsequent exploration of the MCMC chains [27, Ch. 4][18, 19, 12]. Parallelization, although not directly applicable due to the sequencial nature of these algorithms, is also of great interest to the

MCMC community not only for the potential speed-ups but also because it fosters the exploration of the state space. Namely, running several parallel MCMC algorithms can help in discovering important regions of the posterior support [9, Ch. 6]. Population MCMC algorithms work by evolving a population of states, rather than a single one, so the information discovered by the individual chains is shared in order to improve the overall efficiency [27, Ch. 5]. Furthermore, population MCMC algorithms can be combined with independent parallel chains by running "horizontal" and "vertical" steps in the orthogonal MCMC framework [32].

Let us consider N independent parallel MCMC chains, and let $\mathbf{x}_{n,t}$ denote the state of the n-th chain at iteration t. Let $\mathcal{T}_n(\mathbf{x}_{n,t} \to \mathbf{x}_{n,t+1})$ denote the transition kernel applied to the state $\mathbf{x}_{n,t}$ in order to obtain the next state. For simplicity, consider that all $\mathcal{T}_n(\mathbf{x}_{n,t} \to \mathbf{x}_{n,t+1})$ are MH steps with proposal $q(\mathbf{x}|\mathbf{x}_{n,t})$ as described above. More generally, each $\mathcal{T}_n(\mathbf{x}_{n,t} \to \mathbf{x}_{n,t+1})$ could represent the application of one step of a different MCMC algorithm. By applying $\mathcal{T}_n(\mathbf{x}_{n,t} \to \mathbf{x}_{n,t+1})$ to each chain, the population $\mathbf{X}_t = \{\mathbf{x}_{1,t}, \ldots, \mathbf{x}_{N,t}\}$ advances to $\mathbf{X}_{t+1} = \{\mathbf{x}_{1,t+1}, \ldots, \mathbf{x}_{N,t+1}\}$. The whole update can be written as $\mathcal{T}^V(\mathbf{X}_t \to \mathbf{X}_{t+1}) = \prod_{n=1}^{N} \mathcal{T}_n(\mathbf{x}_{n,t} \to \mathbf{x}_{n,t+1})$. Since the MH steps target the posterior $\bar{\pi}(\mathbf{x})$, the population $\mathbf{X}_{t+1}$ is an independent sample from $\bar{\pi}(\mathbf{x})$. Now, consider advancing the population by applying another transition "horizontal" kernel $\mathcal{T}^H(\mathbf{X}_{t+1} \to \mathbf{X}_{t+2})$ that works on the whole population simultaneously. This transition kernel represents the application of one step of a population MCMC algorithm, whose target distribution is also $\bar{\pi}(\mathbf{x})$. The Orthogonal MCMC algorithms alternate between applying these two types of kernels, hence leveraging independent and cooperative exploration [32].

1.2.2. Importance sampling

Importance sampling (IS) consists in rewriting the integral of interest in Eq. (1.5) as a expected value with respect to a pdf $q(\mathbf{x})$, called *importance density* or *proposal*,

$$I = \int_{\mathcal{X}} f(\mathbf{x}) \frac{\bar{\pi}(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} = \mathbb{E}_{q(\mathbf{x})}\left[\frac{\bar{\pi}(\mathbf{x})}{q(\mathbf{x})} f(\mathbf{x})\right]. \tag{1.8}$$

This identity immediately suggests the following standard IS estimator

$$I \approx \widehat{I}_{\text{IS}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\bar{\pi}(\mathbf{z}_i)}{q(\mathbf{z}_i)} f(\mathbf{z}_i), \quad \{\mathbf{z}_i\}_{i=1}^{N} \sim q(\mathbf{x}). \tag{1.9}$$

In comparison with Eq. (1.6), here the evaluations $f(\mathbf{z}_i)$ are weighted according to the ratio $\frac{\bar{\pi}(\mathbf{z}_i)}{q(\mathbf{z}_i)}$, that accounts to the mismatch between target and proposal. However, since it is not possible to evaluate $\bar{\pi}(\mathbf{x})$, but only $\pi(\mathbf{x})$, due to Z being unknown, the above estimator is not used in practice. Let $w_i = \frac{\pi(\mathbf{z}_i)}{q(\mathbf{z}_i)}$ denote the importance weight associated with sample $\mathbf{z}_i$. The following *self-normalized* IS estimator has to be used in practice,

$$I \approx \widehat{I}_{\text{sn-IS}} = \frac{\sum_{i=1}^{N} w_i f(\mathbf{z}_i)}{\sum_{j=1}^{N} w_j} = \sum_{i=1}^{N} \bar{w}_i f(\mathbf{z}_i), \quad \{\mathbf{z}_i\}_{i=1}^{N} \sim q(\mathbf{x}), \tag{1.10}$$

where $\bar{w}_i = \frac{w_i}{\sum_{j=1}^{N} w_j}$. The estimator $\widehat{I}_{\text{sn-IS}}$ is not unbiased but consistent [42].

Compared to MCMC, the IS algorithms have the advantage of using independent samples, which facilitates their theoretical validation. Another advantage is that they provide with an straightforward estimation of the marginal likelihood,

$$Z = \int_{\mathcal{X}} \pi(\mathbf{x}) d\mathbf{x} \,, \tag{1.11}$$

by taking the average of the importance weights,

$$Z \approx \widehat{Z} = \frac{1}{N} \sum_{i=1}^{N} w_i, \tag{1.12}$$

which is an unbiased estimation of Z. However, IS algorithms should not be viewed as a disjunctive alternative to MCMC. In fact, state-of-the-art Monte Carlo methods combine MCMC and IS [36, 35, 31]. This is particularly noticeable in the task of marginal likelihood computation, since MCMC do not provide with an straightforward estimation and so it is often employed in conjunction with IS [36, 35].

A crucial aspect of any IS algorithm is the choice of proposal $q(\mathbf{x})$. Optimal expressions of $q(\mathbf{x})$ can be analytically derived for the estimators $\widehat{I}_{\text{sn-IS}}$ and $\widehat{Z}$. Unsurprisingly, the optimal proposal for $\widehat{Z}$,

$$q_{\text{opt}} = \arg\min_{q} \text{Var}[\widehat{Z}] = \arg\min_{q} \text{Var}\left[\frac{\pi(\mathbf{x})}{q(\mathbf{x})}\right], \tag{1.13}$$

is $q_{\text{opt}}(\mathbf{x}) = \bar{\pi}(\mathbf{x})$, since in this case it is trivial to obtain $\widehat{Z} = Z$. Although implementing this $q_{\text{opt}}(\mathbf{x})$ is not possible in practice, it informs us that $q(\mathbf{x})$ should be chosen so that it is close to $\bar{\pi}(\mathbf{x})$ in order to minimize the variance of the importance weights. Additionally, minimizing the variance of the importance weights is a sensible strategy for obtaining an efficient $\widehat{I}_{\text{sn-IS}}$, as the mean squared error of this estimator scales with the Pearson divergence between posterior and proposal [1]. As a consequence, IS algorithms that iteratively build $q(\mathbf{x})$ have been proposed, going under the name of adaptive IS, that we discuss in the next section.

1.2.3. Adaptive Importance Sampling

The basic idea of adaptive IS (AIS) algorithms is to use the information gained about $\bar{\pi}(\mathbf{x})$ from past weighted samples to improve the proposal. Many AIS algorithms iterate the flow "sample–weigh–adapt" [10]. The proposal $q_t(\mathbf{x})$ has now a subindex that depends on the iteration step. Starting with an initial proposal $q_0(\mathbf{x})$, an AIS algorithm iterates the following block.

- Sample $\{\mathbf{x}_{n,t}\}_{n=1}^{N} \sim q_t(\mathbf{x})$;

- Weigh $w_{n,t} = \frac{\pi(\mathbf{x}_{n,t})}{q_t(\mathbf{x}_{n,t})}$ for $n = 1, \ldots, N$;
- Adapt $q_t(\mathbf{x})$.

Many AIS algorithms rely on updating a population of proposals, rather than a single proposal. The simultaneous use of several proposals is called multiple IS (MIS), and it allows for distinct weighting schemes that differ in computation cost and efficiency [14]. In order to introduce the basics of MIS, let us consider a static population of M proposals, $q_m(\mathbf{x}),\ m = 1, \ldots, M$. Assume we have drawn N samples from each proposal, denoted as $\{\mathbf{x}_{n,m}\}_{n=1}^{N} \sim q_m(\mathbf{x}),\ m = 1, \ldots, M$. Let

$$w(\mathbf{x}_{n,m}) = \frac{\pi(\mathbf{x}_{n,m})}{\Psi(\mathbf{x}_{n,m})}\,, \tag{1.14}$$

denote the weight associated with sample $\mathbf{x}_{n,m}$. The different weighting schemes differ in the choice of $\Psi(\cdot)$. It can be shown that the so-called *full-deterministic mixture* (f-DM) scheme,

$$\Psi_{\text{f-DM}}(\mathbf{x}_{n,m}) = \frac{1}{M}\sum_{j=1}^{M} q_j(\mathbf{x}_{n,m}), \tag{1.15}$$

has lower variance than the standard choice $\Psi_{\text{std}}(\mathbf{x}_{n,m}) = q_{n,m}(\mathbf{x}_{n,m})$ [14]. Note that in $\Psi_{\text{f-DM}}$, the samples are weighted according to the uniform mixture of all proposals, while in Ψ_{std} each sample is only weighted according to the proposal that actually generated it. Intuitively, $\Psi_{\text{f-DM}}$ is more stable than Ψ_{std} since samples from a "bad" proposal can be compensated by the presence of "good" ones, at the expense of an increase in the computation cost. In fact, $\Psi_{\text{f-DM}}$ requires NM^2 proposal evaluations, compared to Ψ_{std} that requires NM proposal evaluations. Frequently, evaluating $\pi(\mathbf{x})$ is the main computation bottleneck, so choosing between $\Psi_{\text{f-DM}}$ and Ψ_{std} has very little impact on the overall computation time.

The *layered* AIS (LAIS) is a class of AIS algorithms that rely on MCMC algorithms to update the proposal $q_t(\mathbf{x})$, hence separating the updating step from the sampling–weighting steps [31]. LAIS take advantage of the exploratory behavior of MCMC algorithms to sample the location parameters of a population of proposals. Let $\{q_{m,t}(\mathbf{x}|\boldsymbol{\mu}_{m,t})\}_{m=1}^{M}$ denote the population of proposals at time t. LAIS applies a MCMC transition kernel to evolve the location parameters $\{\boldsymbol{\mu}_{m,t}\}_{m=1}^{M}$ for T iterations. For instance, in the simplest case, M independent MH algorithms are used. The final set of TM proposals is sampled and the samples are assigned weights according to three possible MIS weighting schemes. Consider for simplicity that exactly one sample is obtained from each proposal, the following *spatial* and *temporal* f-DM denominators are possible,

$$\Psi_{\text{s}}(\mathbf{x}_{m,t}) = \frac{1}{M}\sum_{j=1}^{M} q_{j,t}(\mathbf{x}_{m,t}), \tag{1.16}$$

$$\Psi_{\text{t}}(\mathbf{x}_{m,t}) = \frac{1}{T}\sum_{\tau=1}^{T} q_{m,\tau}(\mathbf{x}_{m,\tau}). \tag{1.17}$$

The LAIS framework allows for different choices in both the upper layer and lower layer, regarding the sampling of the locations parameters $\boldsymbol{\mu}_{m,t}$, and the weighting of the samples $\mathbf{x}_{m,t}$, resulting in algorithms with different efficiency and computation cost.

Instead of adapting a population of parametric proposals, another possibility is the construction of a single proposal that mimics the posterior distribution using techniques of nonparametric regression, that we discuss in Sect. 1.3.

1.2.4. Noisy Monte Carlo

This section discusses Monte Carlo methods in scenarios where $\pi(\mathbf{x})$ cannot evaluated. These scenarios include latent variable models [3], doubly-intractable posteriors [37], and the likelihood-free setting [38, 6]. In some cases, specific algorithms have been designed (see e.g. [37]). A powerful result is the following: if evaluations of $\pi(\mathbf{x})$ are substituted with unbiased estimations, the algorithms remain exact. To illustrate, let us informally introduce the *noisy* versions of the MH algorithm and standard IS..

Let $\widetilde{\pi}(\mathbf{x})$ denote an unbiased estimation of $\pi(\mathbf{x})$ at $\mathbf{x}$, i.e., $\mathbb{E}[\widetilde{\pi}(\mathbf{x})|\mathbf{x}] = \pi(\mathbf{x})$. A noisy MH algorithm is obtained by substituting the exact evaluations $\pi(\mathbf{x}')$, $\pi(\mathbf{x}_t)$ with $\widetilde{\pi}(\mathbf{x}')$, $\widetilde{\pi}(\mathbf{x}_t)$ in Eq. (1.7), namely

$$\widetilde{\alpha}(\mathbf{x}_t, \mathbf{x}') = \min\left\{1, \frac{\widetilde{\pi}(\mathbf{x}')q(\mathbf{x}_t|\mathbf{x}')}{\widetilde{\pi}(\mathbf{x}_t)q(\mathbf{x}'|\mathbf{x}_t)}\right\}. \tag{1.18}$$

The use of this acceptance probability give rise to two algorithms known in the literature as *pseudo-marginal* MH and *Monte Carlo-within-Metropolis*. These algorithms differ in whether $\widetilde{\pi}(\mathbf{x}_t)$ is recomputed at each iteration or reused from past iteration. The pseudo-marginal theorem states that the latter alternative ensures the algorithm is exact . More generally, instead of using unbiased estimations of $\pi(\mathbf{x})$, the noisy Monte Carlo framework in [2] analyzes the discrepancy between the exact MH and noisy algorithms, where the acceptance probability in Eq. (1.7) is substituted with stochastic approximations.

Similarly, let us denote with $\widetilde{w}_i = \frac{\widetilde{\pi}(\mathbf{x}_i)}{q(\mathbf{x}_i)}$ the noisy importance weight obtained by substituting the exact evaluation $\pi(\mathbf{x}_i)$ with the unbiased estimation $\widetilde{\pi}(\mathbf{x}_i)$. The noisy versions of the estimators in Eqs. (1.10) and (1.12) can be shown to converge to the true quantities. The noisy IS is also named *IS squared* and *random-weight IS* in the literature [15, 16, 49].

In both noisy MH and noisy IS, the use of random realizations produces efficiency losses in the final estimators that scale with the noise variance. In the case of noisy IS, the noise contribute with an additional term in the variance of the estimators, and hence change the expression of optimal proposals. Furthermore, obtaining such unbiased estimations is often a costly step. The use of surrogates $\widehat{\pi}(\mathbf{x})$ of $\pi(\mathbf{x})$, built with regression techniques, can help in alleviating these problems.

1.3. Nonparametric approximations

This section deals with the approximation of $\pi(\mathbf{x})$ from a set of evaluations $\pi(\mathbf{x}_1), \ldots, \pi(\mathbf{x}_N)$, at a set of points $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. The set $\mathbf{X}$ is usually referred to as nodes or design points, and they do not necessarily have to be distributed from $\bar{\pi}(\mathbf{x})$. The evaluations of $\pi(\mathbf{x})$ are assumed to be noiseless for simplicity, but they could be corrupted with noise instead. This task is known as the *recovery problem* in the literature of function approximation from scattered data [50, 44]. Finding an approximation $\pi(\mathbf{x})$ from a set of evaluations is also the goal of machine learning regression algorithms, such as $k-$nearest neighbor or Gaussian processes [11, 40]. Below, we give a general description of nonparametric approximations based on combinations of basis functions placed at the design points. Alternatively, one could seek an approximation of some transformation of $\pi(\mathbf{x})$ or some internal part such as the likelihood function or a forward model. The latter approach is very popular in, e.g., the study of physical phenomena by the use of surrogate models of expensive computer simulators [26, 45, 41, 48].

We consider approximations of $\pi(\mathbf{x})$ as linear combinations of basis functions $\varphi_i(\mathbf{x})$, each one centered at a different node $\mathbf{x}_i$,

$$\widehat{\pi}(\mathbf{x}) = \sum_{i=1}^{N} \beta_i \varphi_i(\mathbf{x}), \tag{1.19}$$

where $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_N]^\top$ is the vector of coefficients. Denote with $\mathbf{d} = [\pi(\mathbf{x}_1), \ldots, \pi(\mathbf{x}_N)]^\top$ the vector of evaluations. The coefficients are determined by imposing the interpolation condition, i.e., $\widehat{\pi}(\mathbf{x}_i) = \pi(\mathbf{x}_i)$ for $i = 1, \ldots, N$. Writing it in matrix form gives

$$\mathbf{K}\boldsymbol{\beta} = \mathbf{d} \Rightarrow \boldsymbol{\beta} = \mathbf{K}^{-1}\mathbf{d}, \tag{1.20}$$

where $\mathbf{K}_{ij} = \varphi_j(\mathbf{x}_i)$ for $1 \leq i, j \leq N$. Hence the basis functions must satisfy that $\mathbf{K}$ is invertible. Substituting Eq. (1.20) in Eq. (1.19) results in $\widehat{\pi}(\mathbf{x})$ being a linear combination of the evaluations $\mathbf{d}$. The passing condition can be relaxed by adding σ^2 times the identity matrix to $\mathbf{K}$, hence having $\boldsymbol{\beta} = (\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{d}$. Connections with kernel regression and radial-basis interpolation can be established by imposing properties on the basis functions $\varphi_i(\mathbf{x})$, such as $\varphi_i(\mathbf{x})$ being a proper covariance function (i.e. a kernel) or $\varphi_i(\mathbf{x})$ being a radial-basis function (i.e. a stationary kernel). For instance, if $\varphi_i(\mathbf{x}) \propto e^{-\frac{1}{2h^2}\|\mathbf{x}-\mathbf{x}_i\|^2}$, namely, a Gaussian kernel, then $\widehat{\pi}(\mathbf{x})$ corresponds to the predictive mean function of a Gaussian process with Gaussian kernel [40]. See, e.g., [34, 25] for connections between Gaussian process and kernel methods.

The error in the approximation of $\pi(\mathbf{x})$ by $\widehat{\pi}(\mathbf{x})$ has been studied in the related literatures of function approximation [50]. Clearly, this error depends on the assumed properties of $\pi(\mathbf{x})$, the chosen $\varphi_i(\mathbf{x})$, as well as on the selection of $\mathbf{X}$. Adding more nodes to $\mathbf{X}$ is likely to improve the approximation, but it also requires more evaluations of $\pi(\mathbf{x})$. This trade-off has been addressed in the active learning and experimental design literatures

[39, 46]. Experimental designs aim at finding $\mathbf{X}$ that optimize some utility function. In active learning, the idea is to use the current approximation of $\pi(\mathbf{x})$ to guide the search of new nodes by optimizing acquisition functions that balance exploration of new regions with exploitation of the already-discovered important regions. Indeed, when the view of GP can be adopted, the uncertainty in the approximation can be used to build these acquisition functions [17].

1.3.1. Uses of the interpolant

Here, we discuss different uses of the interpolant $\widehat{\pi}(\mathbf{x})$ within Monte Carlo algorithms. The main reason of employing an approximation to the posterior instead of the true posterior is the promise of computational savings, as it is assumed that the surrogate is much cheaper to evaluate than the exact model. Hence, a first possibility is to apply a Monte Carlo algorithm directly on $\widehat{\pi}(\mathbf{x})$, rather than on $\pi(\mathbf{x})$. For instance, in the MH algorithm this account to using $\widehat{\pi}(\mathbf{x})$ in the computation of the acceptance probability in Eq. (1.7). Similarly, the interpolant $\widehat{\pi}(\mathbf{x})$ can be used to compute the weights in IS algorithms. The resulting Monte Carlo algorithms are approximate, in the sense of the final estimators converging to expectations with respect to (the density proportional to) $\widehat{\pi}(\mathbf{x})$. However, the increase in bias can be compensated with a greater reduction in variance as the algorithms can be run for more iterations in the same computation time. Furthermore, when only unbiased estimations of the posterior are available, the use of a surrogate can be motivated by a denoising effect. A related approach is to substitute $\widehat{\pi}(\mathbf{x})$ directly in the integrals in Eqs. (1.5) and (1.11), giving rise to quadrature formulas [47, 8]. In some cases, the resulting integrals can have closed form. In other cases, intensive Monte Carlo algorithms or other quadratures can be applied as they no longer require the evaluation of $\pi(\mathbf{x})$. These approaches can be used jointly with strategies for improving the interpolant $\widehat{\pi}(\mathbf{x})$ in order to reduce the bias.

A second possibility is employing $\widehat{\pi}(\mathbf{x})$ as proposal within Monte Carlo. The use of interpolative proposals in Monte Carlo has a long history, and can be ascribed to the rejection sampling and adaptive rejection sampling schemes [21, 20, 23]. Informally, a Monte Carlo algorithm can be viewed as a method for generating samples from a target density by "filtering" samples from a proposal. Then, the closer the proposal to the target density, the higher the efficiency. If $\widehat{\pi}(\mathbf{x})$ is employed as independent proposal in a MH algorithm, the acceptance probability in Eq. (1.7) approaches 1 as $\widehat{\pi}(\mathbf{x})$ gets closer to $\pi(\mathbf{x})$. In the case of IS, we already saw in Eq. (1.13) that the optimal proposal (that minimizes the variance of the importance weights) is $q_{\text{opt}}(\mathbf{x}) \propto \pi(\mathbf{x})$. Hence, using $q(\mathbf{x}) \propto \widehat{\pi}(\mathbf{x})$ is a sensible option. However, finding a construction of $\widehat{\pi}(\mathbf{x})$ that is sufficiently close to $\pi(\mathbf{x})$ but is still easy to sample is not trivial. Indeed, the task of sampling $\widehat{\pi}(\mathbf{x})$ can be as hard as the original task of sampling $\pi(\mathbf{x})$. The popular *delayed-acceptance* schemes employ a two-step procedure to deal with this issue [13, 5]. First, a MH step is applied considering $\widehat{\pi}(\mathbf{x})$ as the target density, i.e., substituting $\widehat{\pi}(\mathbf{x})$ instead of $\pi(\mathbf{x})$ in Eq. (1.7). Upon acceptance, another MH

step is applied to correct the algorithm, substituting $\widehat{\pi}(\mathbf{x})$ instead of $q(\mathbf{x}|\mathbf{x}_t)$ and using $\pi(\mathbf{x})$ as the target density in Eq. (1.7). When the proposed state is accepted in the first step, it is considered as it were sampled from $\widehat{\pi}(\mathbf{x})$. This algorithm can be also viewed as way of saving computation time for "bad" proposals, since those that are rejected in the first step are not tested against $\pi(\mathbf{x})$.

1.4. Main contributions

The contributions of this thesis are divided in seven different chapters. The seven chapters can be classified according to the following related topics.

- Chapters 2 and 3 introduce two novel Monte Carlo algorithms based on parallel and population MCMC.
- Chapters 4 and 5 introduce two novel Monte Carlo algorithms that use a nonparametric approximation of the posterior.
- Chapters 6 and 7 study Monte Carlo algorithms in presence of noise in the posterior evaluation.
- Chapters 8 is framed within the theme of Bayesian model selection.

Chapters 2 and 3 deal with the problem of applying Monte Carlo algorithms when the posterior is multimodal or highly concentrated, in which case exploration becomes difficult. Chaper 2 proposes a novel MCMC scheme called *Parallel Metropolis-Hastings Coupler* (PMHC), that combines the use of parallel independent MH algorithms with a population MCMC algorithm called Normal Kernel Coupler (NKC). The use of independent MCMC chains allows to explore different regions, hence avoiding that they all be trapped in the same region. The combination with NKC, a powerful population MCMC algorithm, allows also to share the information discovered by the different chains. Hence, PMHC is especially suitable for multimodal scenarios. Chapter 3 propose *MCMC-driven importance samplers* for Bayesian inference. More specifically, the proposed algorithms are extensions of the LAIS framework for dealing with challenging problems such as concentrated posteriors. Here, the MCMC chains use subsets of data in order to produce a data-tempering effect, which allows to locate faster the regions of high-posterior probability. Since LAIS is an adaptive IS algorithm, the states of the MCMC are not used directly in the estimators, but they serve as location parameters for the subsequent MIS scheme. This also contributes to a more robust overall behavior. Several MCMC algorithms and MIS weighting strategies, as well as recycling and compression schemes to reduce the cost of the approaches, are proposed and tested.

Chapters 4 and 5 both deal with the construction of a nonparametric approximation of

the posterior using regression techniques. The work in Chapter 4 is motivated by posterior densities that are costly to evaluate. Hence, our goal, instead of applying Monte Carlo algorithms directly on the posterior, is to make a more efficient use of the posterior evaluations by a building a surrogate model. The framework proposed in Chapter 4 employs the approximation to build quadratures of posterior expectations and the marginal likelihood. Two types of regression techniques are applied, namely, interpolation with Gaussian kernels and nearest neighbor approximation. These constructions then lead to quadratures that require the use of additional Gaussian quadratures and IS, respectively, but do not depend on evaluating further the posterior. A procedure is proposed for sequentially improving the interpolant by maximizing a suitable acquisition function, composed of two terms to balance exploration and exploitation. The algorithms are tested in simulated examples and a real application of exoplanet detection. On the other hand, Chapter 5 proposes to employ the interpolant as a proposal within an adaptive IS algorithm, named *Regression Adaptive Deep Importance Sampling* (RADIS). The motivation here is to build a very efficient Monte Carlo algorithm by building a proposal that mimics the posterior distribution. Compared to Chapter 4, the resulting algorithm targets the true posterior since the interpolant is employed only as proposal density. The same nonparametric constructions are considered in this work for building the interpolant. The challenge here is to be able to sample from the interpolant. This is achieved by an additional sampling-importance-resampling procedure. The resulting samples are approximately distributed from the interpolant. These samples are weighted according to the posterior and then are used to update the interpolant for the next iteration. RADIS is tested in the retrieval of biophysical parameters of the radiatrive transfer PROSAIL model.

Chapter 6 and 7 consider Monte Carlo algorithms with noisy evaluations of the posterior. Chapter 6 surveys the use of surrogates (i.e. approximations of the posterior built with noisy evaluations) within Monte Carlo algorithms for dealing with noisy and costly posteriors. Hence, Chapter 6 serves as a bridge between the cited scenario and Chapter 5. The main contribution of Chapter 6 is the classification of the studied algorithms in different families, and providing several explanatory tables and figures. Specifically, the Monte Carlo algorithms using surrogates are divided in three broad classes (i) two-stage, (ii) iterative refinement, and (iii) exact. A schematic view of the different families in terms of a series of building blocks is also provided. For instance, RADIS from Chapter 5 is included in the family of exact methods, and it is indeed the IS analogous of the well-known delayed-acceptance MCMC schemes. Chapter 6 also discusses several application scenarios where it is common to work with noisy posterior evaluations, such as Approximate Bayesian Computation (ABC) and reinforcement learning. Chapter 7 focus the study on noisy IS, i.e., IS when the posterior evaluations are corrupted with noise. The expressions for the optimal proposals of the standard, self-normalized and marginal likelihood estimators are derived, which is the main contribution of the work. The expressions of the optimal proposals feature the noise variance, allowing practitioners to account for it when designing IS algorithms in the presence of noise.

Chapter 8 is devoted to Bayesian model selection. Chapter 8 reviews computational approaches for estimating the marginal likelihood of a model, which is the key quantity for performing model comparison. We have classified the approaches in 4 families, namely, (1) Deterministic approximations, (2) Methods based on density estimation, (3) Importance sampling schemes, and (4) Methods based on a vertical representation. Special focus is on IS based approaches, which are the largest family containing the most popular algorithms. The different techniques are presented with a unified notation, highlighting their differences, connections, limitations and strengths. Other aspects such as the use of improper priors and the connection of marginal likelihoods with information criteria and Bayesian predictive model selection are briefly discussed.

Bibliography

[1] Ö. D. Akyildiz and J. Míguez. Convergence rates for optimised adaptive importance samplers. *Statistics and Computing*, 31(2):1–17, 2021.

[2] P. Alquier, N. Friel, R. Everitt, and A. Boland. Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels. *Statistics and Computing*, 26(1-2):29–47, 2016.

[3] C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

[4] C. Andrieu and J. Thoms. A tutorial on adaptive MCMC. *Statistics and computing*, 18(4):343–373, 2008.

[5] M. Banterle, C. Grazian, A. Lee, and C. P. Robert. Accelerating Metropolis–Hastings algorithms by delayed acceptance. *Foundations of Data Science*, 1(2):103, 2019.

[6] M. A. Beaumont. Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41(1):379–406, 2010.

[7] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[8] F.-X. Briol, C. J. Oates, M. Girolami, M. A. Osborne, and D. Sejdinovic. Probabilistic integration: A role in statistical computation? *Statistical Science*, 34(1):1–22, 2019.

[9] S. Brooks, A. Gelman, G. Jones, and X. L. Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall CRC Press, 2011.

[10] Monica F Bugallo, Victor Elvira, Luca Martino, David Luengo, Joaquin Miguez, and Petar M Djuric. Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017.

[11] G. H. Chen and D. Shah. Explaining the success of nearest neighbor methods in prediction. *Foundations and Trends® in Machine Learning*, 10(5-6):337–588, 2018.

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

[13] J. A. Christen and C. Fox. Markov Chain Monte Carlo using an approximation. *Journal of Computational and Graphical statistics*, 14(4):795–810, 2005.

[14] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Generalized Multiple Importance Sampling. *Statistical Science*, 34(1):129–155, 2019.

[15] P. Fearnhead, O. Papaspiliopoulos, and G. O. Roberts. Particle filters for partially observed diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):755–777, 2008.

[16] P. Fearnhead, O. Papaspiliopoulos, G. O. Roberts, and A. Stuart. Random-weight particle filtering of continuous time processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):497–512, 2010.

[17] P. I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[18] C. J. Geyer. Markov chain Monte Carlo maximum likelihood. 1991.

[19] C. J. Geyer and E. A. Thompson. Annealing Markov Chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90:909–920, 1995.

[20] W. R. Gilks, N. G. Best, and K. K. C. Tan. Adaptive Rejection Metropolis Sampling within Gibbs Sampling. *Applied Statistics*, 44(4):455–472, 1995.

[21] W. R. Gilks and P. Wild. Adaptive Rejection Sampling for Gibbs Sampling. *Applied Statistics*, 41(2):337–348, 1992.

[22] Mark Girolami and Ben Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

[23] D. Görür and Y. W. Teh. Concave convex adaptive rejection sampling. *Journal of Computational and Graphical Statistics (to appear)*, 2010.

[24] Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, April 2001.

[25] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

[26] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[27] F. Liang, C. Liu, and R. Caroll. *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*. Wiley Series in Computational Statistics, England, 2010.

[28] D. J. C. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[29] L. Martino. A review of multiple try MCMC algorithms for signal processing. *Digital Signal Processing*, 75:134–152, 2018.

[30] L. Martino, R. Casarin, F. Leisen, and D. Luengo. Adaptive independent sticky MCMC algorithms. *EURASIP Journal on Advances in Signal Processing (to paper)*, 2017.

[31] L. Martino, V. Elvira, D. Luengo, and J. Corander. Layered adaptive importance sampling. *Statistics and Computing*, 27(3):599–623, 2017.

[32] L. Martino, V. Elvira, D. Luengo, J. Corander, and F. Louzada. Orthogonal parallel MCMC methods for sampling and optimization. *Digital Signal Processing*, 58(Supplement C):64 – 84, 2016.

[33] L. Martino, D. Luengo, and J. Míguez. Independent random sampling methods. *Springer*, 2018.

[34] L. Martino and J. Read. A joint introduction to Gaussian Processes and Relevance Vector Machines with connections to Kalman filtering and other kernel smoothers. *Information Fusion*, 74:17–38, 2021.

[35] P. Del Moral, Arnaud Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[36] R. M. Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

[37] J. Park and M. Haran. Bayesian inference in the presence of intractable normalizing functions. *Journal of the American Statistical Association*, 113(523):1372–1390, 2018.

[38] L. F. Price, C. C. Drovandi, A. Lee, and D. J. Nott. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1):1–11, 2018.

[39] L. Pronzato and W. G. Müller. Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22(3):681–701, 2012.

[40] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[41] Saman Razavi, Bryan A Tolson, and Donald H Burn. Review of surrogate modeling in water resources. *Water Resources Research*, 48(7), 2012.

[42] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[43] C. P. Robert, V. Elvira, N. Tawn, and C. Wu. Accelerating MCMC algorithms. *Wiley Interdisciplinary Reviews: Computational Statistics*, 10(5):e1435, 2018.

[44] R. Schaback. *Reconstruction of multivariate functions from scattered data*. PhD thesis, Citeseer, 1997.

[45] Claude J Schmit and Jonathan R Pritchard. Emulation of reionization simulations for bayesian inference of astrophysics parameters using neural networks. *Monthly Notices of the Royal Astronomical Society*, 475(1):1213–1223, 2018.

[46] B. Settles. Active learning literature survey. 2009.

[47] A. Sommariva and M. Vianello. Numerical cubature on scattered data by radial basis functions. *Computing*, 76(3-4):295, 2006.

[48] D. H. Svendsen, L. Martino, and G. Camps-Valls. Active emulation of computer codes with gaussian processes - application to remote sensing. *Pattern Recognition*, 100:107103, 2020.

[49] M.-N. Tran, M. Scharth, M. K. Pitt, and R. Kohn. Importance sampling squared for Bayesian inference in latent variable models. *arXiv preprint arXiv:1309.3339*, 2013.

[50] H. Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004.

2. PARALLEL METROPOLIS-HASTINGS COUPLER

In *IEEE Signal Processing Letters*, Volume 26, 953–957 (2019)

Fernando Llorente*, Luca Martino*, David Delgado*
* Universidad Carlos III de Madrid, Léganes, Madrid, Spain

Abstract

Bayesian methods and their implementations by means of sophisticated Monte Carlo (MC) techniques, such as Markov chain Monte Carlo (MCMC) and particle filters, have become very popular in signal processing over the last years. In this work, we present a novel interacting parallel MCMC scheme, called Parallel Metropolis-Hastings Coupler (PMHC), where the information provided by different parallel MCMC chains is properly combined by the use of another advanced MCMC method, called Normal Kernel Coupler (NKC). The NKC employs a mixture of densities as proposal density, which is updated according to a population of states. The PMHC is particularly efficient in multimodal scenarios since it obtains a faster exploration of the state space with respect to other benchmark techniques. Several numerical simulations are provided showing the efficiency and robustness of the proposed method.

Keywords: Bayesian Inference, MCMC algorithms, Normal Kernel Coupler, parallel MCMC, population MCMC.

2.1. Introduction

The Markov Chain Monte Carlo (MCMC) are well-known Monte Carlo (MC) methodologies, which have become very popular in signal processing, statistics and machine learning during the past decades, in order to perform Bayesian inference and stochastic optimization [1, 5, 6, 14]. They generate a Markov chain which converges to the desired stationary probability density function (pdf).

In the last decades, the available computing power has also grown substantially. For instance, the possible use of a network of processors/machines has become common and widespread. For this reason, the interest in running several independent parallel MCMC chains has also increased in the last years. Clearly, If N chains of length T are run in parallel processors, the total number of generated samples is NT instead of N, obtained with the same running time of a unique chain. Even if only one processor is used, employing independent parallel chains (IPCs) speeds up the exploration of the state space, which is crucial for the performance of a MCMC technique, specially in high-dimensional applications [4, 12, 13, 7]. Thus, several MC schemes consider the application of parallel chains

for building efficient samplers, even when only one processor is employed [12, 13, 3, 2]. For instance, in [4, 8] , several chains are run in parallel, and a unique joint proposal or different proposal densities are adapted using the samples generated by all the chains. In [15, 10] a modified target pdf is considered to produce a repulsion among the parallel chains. In [11], the authors design an importance sampling scheme, where the IPCs are used for adapting the location parameters of different proposal densities.

In this work, we present a novel MCMC scheme, called *Parallel Metropolis-Hastings Coupler* (PMHC). The PMHC combines efficiently the use of the IPCs with a population-based MCMC technique, called Normal Kernel Coupler (NKC) [16]. The PMHC obtains a fast exploration of the state space, ensuring the overall ergodicity. The information provided by N parallel MCMC chains are properly shared by the use of the NKC. The NKC is an advanced MCMC technique which uses a mixture of densities as proposal pdf. This mixture is updated using the information given by a cloud of generated states. Specifically, NKC induces a suitable random walk over a population of states which determines the means of components of the proposal mixture, ensuring the ergodicity [16].

When it is run in a unique processor, the PMHC can be considered a population MCMC scheme, which is particularly efficient in highly multimodal scenarios. When several independent processors are also employed, the computational speed up, as result of the parallelization of the IPCs, is considered an additional advantage of the proposed approach.

More specifically, the PMHC belongs to the class of Orthogonal MCMC algorithms (OMCMC) [12], formed by vertical and horizontal iterations. The vertical iterations, i.e., the IPCs, and the horizontal iterations, i.e., the NKC scheme, are cyclically repeated until reaching the desired number of samples. Unlike the OMCMC schemes proposed in [12], the PMHC employs a random walk kernel in the horizontal steps, due to the application of the NKC. This explorative behavior in the horizontal steps increases the robustness and the efficiency of the resulting algorithm, as shown in the numerical simulations.

The remaining of the paper is organized as follows. The problem statement is described in Section 2.2. The proposed technique is introduced in Section 2.3. Section 2.4 contains the numerical experiments. We conclude with a brief summary in Section 2.5.

2.2. Bayesian inference

In many applications, we aim at inferring a variable of interest given a set of observations or measurements. Let us denote the variable of interest by $\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^{d_x}$, and let $\mathbf{y} \in \mathbb{R}^{d_y}$ be the observed data. The posterior pdf is then

$$\bar{\pi}(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{Z(\mathbf{y})}, \tag{2.1}$$

where $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf and $Z(\mathbf{y})$ is the model evidence (a.k.a. marginal likelihood). Generally, $Z(\mathbf{y})$ is unknown, so we are able to evaluate the

unnormalized target function,

$$\pi(\mathbf{x}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}). \tag{2.2}$$

The analytical study of the posterior density $\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x})$ is unfeasible, so that numerical approximations are required. Our goal is to approximate efficiently the measure of $\bar{\pi}(\mathbf{x})$ employing a cloud of random samples. Thus, an integral involving $\bar{\pi}(\mathbf{x})$ can be approximated via Monte Carlo quadrature. Generally, a direct method for drawing independent samples from $\bar{\pi}(\mathbf{x})$ is not available and alternative approaches (e.g., MCMC algorithms) are needed. The only required assumption is being able to evaluate the unnormalized target function $\pi(\mathbf{x})$.

2.3. Parallel Metropolis-Hastings Coupler

In this section we describe the Parallel Metropolis-Hastings Coupler (PMHC) scheme. At the t-th iteration, with $t \in \mathbb{N}$, the PMHC algorithm considers a population of samples

$$\mathcal{P}_t = \{\mathbf{x}_{1,t}, \mathbf{x}_{2,t}, \dots, \mathbf{x}_{N,t}\} = \{\mathbf{x}_{n,t}\}_{n=1}^{N}. \tag{2.3}$$

In a first stage of the algorithm, N independent parallel MH chains are run considering $\mathcal{P}_t = \{\mathbf{x}_{n,t}\}_{n=1}^{N}$ as current states. Each chain performs T_V (*vertical*) iterations, yielding a new population $\mathcal{P}_{t+T_V} = \{\mathbf{x}_{n,t+T_V}\}_{n=1}^{N}$ of samples. Then, the information of the parallel independent chains is mixed by the use of T_H (*horizontal*) iterations of the Normal Kernel Coupler (NKC) technique. The NKC employs a mixture of densities $\varphi(\mathbf{x}|\boldsymbol{\mu}_{n,t}, \boldsymbol{\Sigma}_n)$ as proposal density $\Psi(\mathbf{x}|\mathcal{P}_t)$ (where $\boldsymbol{\mu}_{n,t}$ and $\boldsymbol{\Sigma}_n$ represent a mean and a covariance matrix, respectively). In the horizontal iterations, the current cloud of samples $\mathcal{P}_t$ are used as means, $\boldsymbol{\mu}_{n,t}$, of the N components $\varphi(\mathbf{x}|\boldsymbol{\mu}_{n,t}, \boldsymbol{\Sigma}_n)$. The NKC generates a random walk behavior, where the mixture $\Psi(\mathbf{x}|\mathcal{P}_t)$ changes with the iterations, considering only the last state of chain. After T_H iterations of the NKC, one *epoch* of the PMHC has been completed. Another epoch of the PMHC consists of running again T_V iterations of the N parallel MH chains using as initial states the samples contained in the current population $\mathcal{P}_t$, and then performing again the T_H of the NKC. The total number of epochs is denoted as M.

The general outline of the PMHC algorithm is given in Table 2.1. The standard MH algorithm used in a vertical chain is described in Table 2.2. The NKC is summarized in Table 2.3 and also described in Section 2.3.1. It is possible to ensure the ergodicity of the PMHC since its kernel function can be expressed as product of kernel functions with invariant density $\bar{\pi}(\mathbf{x})$ (for further details, see [12]).

2.3.1. Normal Kernel Coupler (NKC)

Let us denote with $\tau \in \mathbb{N}$ another iteration index, in order to avoid confusion with the global index t of the PMHC. At iteration τ, given the current population of samples $\mathcal{P}_{\tau-1} =$

Table 2.1: Parallel Metropolis-Hastings Coupler

- **Initialization:** Choose the N starting vectors, $\mathcal{P}_0 = \{\mathbf{x}_{n,0}\}_{n=1}^N$, three values $M, T_V, T_H \in \mathbb{N}^+$, and set $t = 0$,
- **For** $m = 1, \ldots, M$**:**

1. **Parallel MH steps:** Using the cloud $\mathcal{P}_t$ as initial states, perform T_V iterations of N parallel MH schemes, obtaining the populations $\mathcal{P}_{t+1}, \mathcal{P}_{t+2} \ldots \mathcal{P}_{t+T_V}$.
2. **NKC steps:** Using the cloud $\mathcal{P}_{t+T_V}$, perform T_H iterations of the NKC, obtaining the populations $\mathcal{P}_{t+T_V+1}, \mathcal{P}_{t+T_V+2} \ldots \mathcal{P}_{t+T_V+T_H}$.
3. Set $t \leftarrow t + T_V + T_H$.

- **Outputs:** The $NT = NM(T_V + T_H)$ samples contained in $\{\mathcal{P}_t\}_{t=1}^T$ with $T = M(T_V + T_H)$.

Table 2.2: n-th Metropolis-Hastings (MH) chain

- **Initialization:** Choose the initial state, $\mathbf{x}_{n,0}$, and $T_V \in \mathbb{N}$.
- **For** $\tau = 1, \ldots, T_V$**:**

1. Draw $\mathbf{x}' \sim q_n(\mathbf{x}|\mathbf{x}_{n,\tau-1})$.
2. Set $\mathbf{x}_{n,\tau} = \mathbf{x}'$ with probability

$$\alpha = \min\left[1, \frac{\pi(\mathbf{x}')q(\mathbf{x}_{n,\tau-1}|\mathbf{x}')}{\pi(\mathbf{x}_{n,\tau-1})q(\mathbf{x}'|\mathbf{x}_{n,\tau-1})}\right], \tag{2.4}$$

otherwise, set $\mathbf{x}_{n,\tau} = \mathbf{x}_{n,\tau-1}$ (with probability $1 - \alpha$).

- **Outputs:** The T_V samples $\{\mathbf{x}_{n,\tau}\}_{\tau=1}^{T_V}$.

$\{\mathbf{x}_{n,\tau-1}\}_{n=1}^N$, the NKC [16] employs the following mixture

$$\Psi(\mathbf{x}|\mathcal{P}_{\tau-1}) = \frac{1}{N}\sum_{n=1}^{N} \varphi(\mathbf{x}|\mathbf{x}_{n,\tau-1}, \sigma_h^2\mathbf{I}), \tag{2.5}$$

as proposal density in a MH-type algorithm. Note that $\varphi(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a density with mean $\boldsymbol{\mu}$, covariance matrix $\boldsymbol{\Sigma}$; $\sigma_h > 0$ is chosen by the user and $\mathbf{I}$ is a $d_x \times d_x$ identity matrix.

The NKC scheme can be seen as MH-type method where: **(a)** a "past" sample $\mathbf{x}_{k,\tau-1}$ is selected uniformly within $\mathcal{P}_{\tau-1}$, to compare it with a new candidate $\mathbf{x}'$ as new state of

the chain; **(b)** a new candidate $\mathbf{x}'$ is drawn from the mixture $\Psi(\mathbf{x}|\mathcal{P}_{\tau-1})$; **(c)** then a test involving $\mathbf{x}_{k,\tau-1}$ and $\mathbf{x}'$ is performed, in order to choose the next state $\mathbf{x}_{k,\tau}$ of the chain; more precisely, with a suitable probability α shown Eq. (2.7), we set $\mathbf{x}_{k,\tau} = \mathbf{x}'$; otherwise, with probability $1-\alpha$, we set $\mathbf{x}_{k,\tau} = \mathbf{x}_{k,\tau-1}$; **(d)** we update the population $\mathcal{P}_\tau = \{\mathbf{x}_{n,\tau}\}_{n=1}^N$. Note that the new population, $\mathcal{P}_\tau$, differs to the previous one, $\mathcal{P}_{\tau-1}$, at most for one sample. Namely, if the new candidate has been accepted, i.e., $\mathbf{x}_{k,\tau} = \mathbf{x}'$, then $\mathcal{P}_\tau$ differs to $\mathcal{P}_{\tau-1}$ in the k-th element. Otherwise, if $\mathbf{x}_{k,\tau} = \mathbf{x}_{k,\tau-1}$, then we have $\mathcal{P}_\tau = \mathcal{P}_{\tau-1}$.

Since the population $\mathcal{P}_\tau$ varies with τ, then the proposal mixture $\Psi(\mathbf{x}|\mathcal{P}_\tau)$ is also changing with τ. For this reason, several authors have classified the NKC as an adaptive MCMC scheme. More precisely, the NKC is a MH-type algorithm with a random walk proposal density. Indeed, only the last state of the chain is employed to update the proposal. The difference with respect to (w.r.t.) the standard MH method is that the random walk is carried out in a mixture of densities, in a suitable way such that the ergodicity is ensured. Indeed, the NKC can be interpreted as a MH-within-Gibbs scheme working in an extended space with a generalized target $\bar{\pi}(\mathbf{x}_{1:N}) = \prod_{n=1}^N \bar{\pi}(\mathbf{x}_n)$ [16]. Unlike for the adaptive MCMC techniques, no additional theoretical requirements are needed. Table 2.3 provides a detailed description of the NKC, where we have rewritten the mixture in Eq. (2.5) as

$$\begin{aligned}\Psi(\mathbf{x}|\mathcal{P}_{\tau-1}) &= \Psi(\mathbf{x}|\mathcal{P}_{\tau-1}^{(-k)}, \mathbf{x}_{k,\tau-1}) \\ &= \frac{1}{N}\sum_{n\neq k}^{N} \varphi(\mathbf{x}|\mathbf{x}_{n,\tau-1}, \sigma_h^2\mathbf{I}) + \frac{1}{N}\varphi(\mathbf{x}|\mathbf{x}_{k,\tau-1}, \sigma_h^2\mathbf{I}). \end{aligned} \tag{2.6}$$

where $\mathcal{P}_{\tau-1}^{(-k)} = \mathcal{P}_{\tau-1}\backslash\{\mathbf{x}_{k,\tau-1}\}$. Note that equations (2.5) and (2.6) are the same. However, the expression above is required to properly understand the Eq. (2.7) below. The name with the attribute "Normal" is due to the use of Gaussian components, $\varphi(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, in the original work [16].

Now, let us consider the PMHC. The use of the IPCs for generating the initial population of states allows the construction of a tailored proposal mixture that can be interpreted as a kernel density estimation of the target $\bar{\pi}$. Moreover, the NKC is an excellent tool to share the information of different parallel chains as we show in the numerical results.

2.3.2. Computational cost and running time

In the Monte Carlo techniques, the most costly step is often the evaluation of the target posterior density $\bar{\pi}$, due to the use of complex models and/or a large number of data to be analyzed (i.e., a costly likelihood). Generally the other factors, such as sampling from the proposal or performing the MH test, are negligible w.r.t. the cost of evaluating the posterior. Note that in the MH and the NKC methods at each iteration we have just one new evaluation of the target at the novel candidate $\mathbf{x}'$. Therefore, in each epoch of the PMHC, we have $NT_V + T_H$ evaluations of $\bar{\pi}(\mathbf{x})$. Hence, the total number of target

Table 2.3: Normal Kernel Coupler (NKC)

- **Initialization:** Denote as $\mathcal{P}_0 = \{\mathbf{x}_{n,0}\}_{n=1}^N$ the starting cloud of states, and choose the variance, σ_h^2, of each component of the mixture.
- **For** $\tau = 1, \ldots, T_H$**:**

1. Draw uniformly an index $k \in \{1, ..., N\}$.
2. Draw $\mathbf{x}'$ from $\Psi(\mathbf{x}|\mathcal{P}_{\tau-1}) = \Psi(\mathbf{x}|\mathcal{P}_{\tau-1}^{(-k)}, \mathbf{x}_{k,\tau-1})$, given in Eq. (2.6).
3. Set $\mathbf{x}_{k,\tau} = \mathbf{x}'$ with probability

$$\alpha = \min\left[1, \frac{\pi(\mathbf{x}')\Psi(\mathbf{x}_{k,\tau-1}|\mathcal{P}_{\tau-1}^{(-k)}, \mathbf{x}')}{\pi(\mathbf{x}_{k,\tau-1})\Psi(\mathbf{x}'|\mathcal{P}_{\tau-1}^{(-k)}, \mathbf{x}_{k,\tau-1})}\right], \tag{2.7}$$

   otherwise, set $\mathbf{x}_{k,\tau} = \mathbf{x}_{k,\tau-1}$ with probability $1 - \alpha$.
4. Set $\mathbf{x}_{n,\tau} = \mathbf{x}_{n,\tau-1}$ for all $n \neq k$, and $\mathcal{P}_\tau = \{\mathbf{x}_{n,\tau}\}_{n=1}^N$.

- **Outputs:** The NT_H samples $\{\mathcal{P}_\tau\}_{\tau=1}^{T_H}$.

evaluations is $E = M(NT_V + T_H)$ [1].
In terms of running time, i.e., the length of time required to perform a run of the PMHC, we have to take into account that the vertical iterations in one epoch can be parallelized. Therefore, denoting as $s_u = 1$ the unit of time required for performing one iteration of an MCMC algorithm,[2] we have that the total amount of time is $T = M(T_V + T_H)$, where the factor N disappears, w.r.t. E, due to the parallelization. Taking into account these two quantities, E and T, is essential for providing a fair comparison w.r.t. other MCMC algorithms.

2.4. Numerical Simulations

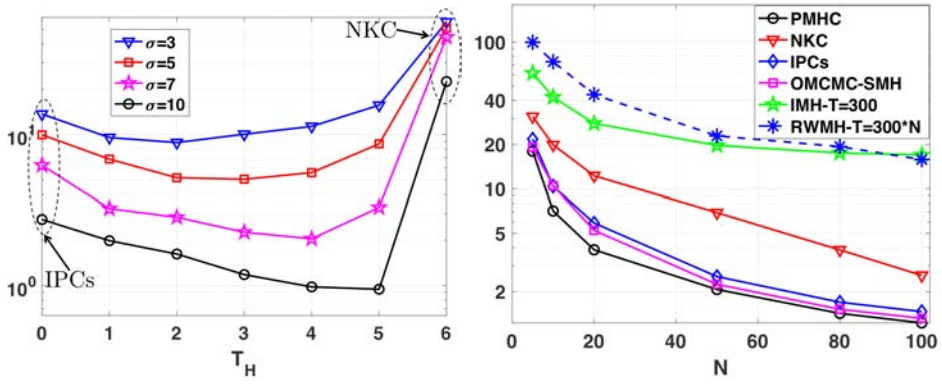
In this section, we test the performance of proposed PMHC technique comparing with other benchmark schemes. We consider a bivariate and highly multimodal target pdf, which is a mixture of 5 Gaussian densities, i.e.,

$$\bar{\pi}(\mathbf{x}) = \frac{1}{5}\sum_{i=1}^{5} \mathcal{N}(\mathbf{x}; \boldsymbol{\nu}_i, \boldsymbol{\Lambda}_i), \quad \mathbf{x} \in \mathbb{R}^2, \tag{2.8}$$

[1] Note that we assume that communication cost among the parallel processors/workers is negligible. We plan to relax this assumption in future works (following, e.g., [9]).

[2] We assume that s_u is the same for each processors. We also consider negligible the difference of running a MH or NHC sampler, since they address the same posterior pdf.

with means $\boldsymbol{\nu}_1 = [-10, -10]^\top$, $\boldsymbol{\nu}_2 = [0, 16]^\top$, $\boldsymbol{\nu}_3 = [13, 8]^\top$, $\boldsymbol{\nu}_4 = [-9, 7]^\top$, and $\boldsymbol{\nu}_5 = [14, -14]^\top$, and with covariance matrices $\boldsymbol{\Lambda}_1 = [2,\ 0.6; 0.6,\ 1]$, $\boldsymbol{\Lambda}_2 = [2,\ -0.4; -0.4,\ 2]$, $\boldsymbol{\Lambda}_3 = [2,\ 0.8; 0.8,\ 2]$, $\boldsymbol{\Lambda}_4 = [3,\ 0; 0,\ 0.5]$, and $\boldsymbol{\Lambda}_5 = [2,\ -0.1; -0.1,\ 2]$. Note that the target pdf $\bar{\pi}(\mathbf{x})$ has 5 different modes. We apply different MCMC algorithms to estimate the expected value $E[\mathbf{X}]$ of the random variable $\mathbf{X} \sim \bar{\pi}(\mathbf{x})$. The ground-truth is $[1.6, 1.4]^\top$. We compute the Mean Square Error (MSE) averaging the results over 500 independent runs. For the vertical MH chains, we consider Gaussian proposals $q_n(\mathbf{x}|\mathbf{x}_{n,t-1}) = \mathcal{N}(\mathbf{x}|\mathbf{x}_{n,t-1}, \sigma_v^2\mathbf{I})$, using the same isotropic covariance matrix. For the horizontal NKC iterations, we employ also Gaussian components for the mixture $\Psi(\mathbf{x})$, i.e., $\varphi(\mathbf{x}|\mathbf{x}_{n,t-1}) = \mathcal{N}(\mathbf{x}|\mathbf{x}_{n,t-1}, \sigma_h^2\mathbf{I})$. Moreover, for the sake of simplicity, we set σ_v and σ_h to a common value denoted as σ, i.e., $\sigma_v = \sigma_h = \sigma$.



(a) (b)

Figure 2.1: MSE in a log-scale as function **(a)** as function of T_H and $T_V = 6 - T_H$ (first experiment; each curve corresponds to a different value of σ), or **(b)** as function of N (second experiment; each curve corresponds to a different MCMC method with $\sigma = 5$).

First Experiment. For a first numerical test, we set $N = 10$, we fix the total number of target evaluations $E = 10^4$, and we consider different values of $\sigma \in \{3, 5, 7, 10\}$. We also set that $T_V + T_H = 6$ and $T_H \in \{0, 1, 2, 3, 4, 5, 6\}$, hence in each epoch we do 6 iterations where T_H are the horizontal NKC steps, and $T_V = 6 - T_H$ the vertical steps. Note that when $T_H = 0$, we only perform the independent parallel chains (IPCs). Otherwise, when $T_H = 6$, we have only NKC steps. Therefore, we can compare the PMHC, the IPCs and the NKC schemes. Since we have fixed $E = 10^4$, the number of epochs is $M = \lfloor \frac{E}{NT_V + T_H} \rfloor$. Furthermore, we choose deliberately an inappropriate initialization to test the robustness of the different schemes (we can observe the exploration abilities of the different methods). Indeed, the initialization does not contain the modes of $\bar{\pi}(\mathbf{x})$. More specifically, we set $\mathbf{x}_{n,0} \sim \mathcal{U}([-4, 4] \times [-4, 4])$, for all $n \in \{1, \ldots, N\}$ and $\mathcal{P}_0 = \{\mathbf{x}_{n,0}\}_{n=1}^N$.

Discussion. The results in terms of MSE are given in Figure 2.1(a). Each curve corresponds to a value of $\sigma \in \{3, 5, 7, 10\}$. All the points at $T_H = 0$ corresponds to the IPCs, and all the points for $T_H = 6$ represents the MSE of the NKC. All the intermediate points, for $1 \leq T_H \leq 5$, corresponds to the PMHC algorithms. We can observe that the PMHC always outperforms the IPCs and NKC for any value of σ and T_H, i.e., regardless the

choices of σ and T_H. Given the same target evaluations E (i.e., a fair comparison), these results clearly show that the PMHC improves the performance of the IPCs and the NKC. Namely, the PMHC is a clever and efficient mix of the IPCs and the NKC kernels (the product of these kernels) that improves the performance of each single method (i.e., IPCs and NKC) applied independently. Note that, in this highly multimodal scenario, the NKC has worse performance compared with the the IPCs. Clearly, this is because the IPCs are able to discover all the modes in a faster way. After this observation, it is surprising to note that the best results of the PMHC with $\sigma = 10$ are obtained for $T_H = 5$ and $T_V = 6 - T_H = 1$. Again, this shows the potentiality of the PMHC. Indeed, only changing one iteration of NKC with one iteration of the IPCs (i.e., $T_H = 5$ and $T_V = 1$, keeping the target evaluations E constant), provides a remarkable drop in the MSE. Notice that the optimal number of horizontal iterations (T_H) increases as σ grows. Indeed, the increase of σ allows the IPC's to explore the state space in a easier way (less iterations are required). This simple example shows that the NKC is an excellent tool for sharing information among the IPCs. At the same time, the IPCs are also excellent tools for improving the performance of the NKC.

Second Experiment. In this case, we set $\sigma = 5$, $M = 50$ and vary $N \in \{5, 10, 20, 50, 80, 100\}$. We compare the PMHC with different MCMC schemes: the IPCs, the NKC, the OMCMC-SMH algorithm in [12], and two single MH methods with a longer chain (one with an independent proposal $\Psi(\mathbf{x}|\mathcal{P}_0)$ and $T = 300$, denoted as IMH, and a second one with a random walk proposal with $T = 300N$, denoted as RWMH). Here, we compare the different methods considering the same running time, $T = M(T_V + T_H) = 300$, and the use of N different parallel processors. Thus, since we set $T_H = T_V = 3$ for the PMHC, we have $T_V = 6$ for the IPCs (and $T_H = 0$) and we have $T_H = 6$ for NKC (and $T_V = 0$). For the OMCMC-SMH algorithm, we also set $T_H = T_V = 3$ and the same proposals q_n and φ, for providing a fair comparison with the PMHC. For all these techniques, we have $T = 300$. For the same reason, we set $T = 300$ for the single Independent MH (IMH) chain with proposal $\Psi(\mathbf{x}|\mathcal{P}_0)$. Note that, when N varies the only change is the number of component in the proposal mixture $\Psi(\mathbf{x}|\mathcal{P}_0)$. Here, we consider a good initialization, setting $\mathbf{x}_{n,0} \sim \mathcal{U}([-20, 20] \times [-20, 20])$ for all n, and $\mathcal{P}_0 = \{\mathbf{x}_{n,0}\}_{n=1}^N$, which covers all the modes of $\bar{\pi}(\mathbf{x})$.

Discussion. The results in terms of MSE are shown in Figure 2.1(b). Again, the PMHC outperforms the rest of methodologies, providing the smallest MSEs. Comparing the results of the NKC and the longer MH chain with a static proposal mixture, we can see the importance of moving the components of the mixture $\Psi(\mathbf{x}|\mathcal{P}_0)$ according to the suitable random walk induced by NKC. The use of the IPCs is even more convenient with the good initialization and with an increasing number of chains. Again, this is owing to the ability to reach quickly all the modes of the target $\bar{\pi}(\mathbf{x})$. The PMHC also outperforms the OMCMC-SMH algorithm, where the Sample Metropolis Hastings (SMH) with an adaptive proposal pdf is employed in the horizontal steps. This confirms that the NKC fits particularly well within the OMCMC scheme as horizontal technique. The benefit of the PMHC with respect to OMCMC-SMH is more evident for small values of N. When N

grows, the performance of the IPCs, OMCMC-SMH and the PMHC becomes more similar due to the good initialization considered in this experiment. Indeed, as N grows, the population-based methods discover in a easier way all the modes of $\bar{\pi}(\mathbf{x})$.

2.5. Conclusions

In this work, we have introduced the Parallel Metropolis-Hastings Coupler (PMHC). The PMHC combines the use of the IPCs with the application of the NKC. The numerical results show that the PMHC outperforms the use of IPCs and NKC applied separately, as well as other benchmark techniques. The PMHC fits particularly well in highly multimodal scenarios, since it obtains a faster exploration of the state space. If several independent processors are available, the IPCs steps in the PMHC can be parallelized, and the computational speed up is an additional advantage of the proposed approach.

Bibliography

[1] J. Candy. *Bayesian signal processing: classical, modern and particle filtering methods*. John Wiley & Sons, England, 2009.

[2] J. Corander, M. Ekdahl, and T. Koski. Parallel interacting MCMC for learning of topologies of graphical models. *Data Mining and Knowledge Discovery*, 17(3):431–456, 2008.

[3] J. Corander, M. Gyllenberg, and T. Koski. Bayesian model learning based on a parallel MCMC strategy. *Statistics Computing*, 16:355–362, 2006.

[4] R. Craiu, J. Rosenthal, and C. Yang. Learn from thy neighbor: Parallel-chain and regional adaptive MCMC. *Journal of the American Statistical Association*, 104(448):1454–1466, 2009.

[5] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Heretical multiple importance sampling. *IEEE Signal Processing Letters*, 23(10):1474–1478, 2016.

[6] W. J. Fitzgerald. Markov chain Monte Carlo methods with applications to signal processing. *Signal Processing*, 81(1):3–18, January 2001.

[7] P. Jacob, C. P. Robert, and M. H. Smith. Using parallel computation to improve Independent Metropolis-Hastings based estimation. *Journal of Computational and Graphical Statistics*, 3(20):616–635, 2011.

[8] J. M. Keith, D. P. Kroese, and G. Y. Sofronov. Adaptive independence samplers. *Statistics and Computing*, 18(4):409–420, 2008.

[9] L. Martino and V. Elvira. Compressed Monte Carlo for distributed Bayesian inference. *viXra:1811.0505*, pages 1–14, 2018.

[10] L. Martino, V. Elvira, D. Luengo, A. Artes, and J. Corander. Smelly parallel MCMC chains. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015.

[11] L. Martino, V. Elvira, D. Luengo, and J. Corander. Layered adaptive importance sampling. *Statistics and Computing*, 27(3):599–623, 2017.

[12] L. Martino, V. Elvira, D. Luengo, J. Corander, and F. Louzada. Orthogonal parallel MCMC methods for sampling and optimization. *Digital Signal Processing*, 58(Supplement C):64 – 84, 2016.

[13] P. Del Moral, Arnaud Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[14] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[15] C. P. Robert and K. L. Mengersen. Iid sampling with self-avoiding particle filters: The pinball sampler, 2001.

[16] G. R. Warnes. The Normal Kernel Coupler: An adaptive Markov Chain Monte Carlo method for efficiently sampling from multi-modal distributions. *Technical Report*, 2001.

3. MCMC-DRIVEN IMPORTANCE SAMPLERS

In *Applied Mathematical Modelling*, Volume 111, 310–331 (2022)

F. Llorente[+], E. Curbelo[+], L. Martino[†], V. Elvira[‡], D. Delgado[+]
[+] Universidad Carlos III de Madrid (UC3M), Spain
[†] Universidad Rey Juan Carlos (URJC), Spain
[‡] University of Edinburgh, UK

Abstract

Monte Carlo sampling methods are the standard procedure for approximating complicated integrals of multidimensional posterior distributions in Bayesian inference. In this work, we focus on the class of layered adaptive importance sampling algorithms, which is a family of adaptive importance samplers where Markov chain Monte Carlo algorithms are employed to *drive* an underlying multiple importance sampling scheme. The modular nature of the layered adaptive importance sampling scheme allows for different possible implementations, yielding a variety of different performances and computational costs. In this work, we propose different enhancements of the classical layered adaptive importance sampling setting in order to increase the efficiency and reduce the computational cost, of both upper and lower layers. The different variants address computational challenges arising in real-world applications, for instance with highly concentrated posterior distributions. Furthermore, we introduce different strategies for designing cheaper schemes, for instance, recycling samples generated in the upper layer and using them in the final estimators in the lower layer. Different numerical experiments show the benefits of the proposed schemes, comparing with benchmark methods presented in the literature, and in several challenging scenarios.

Keywords: Bayesian inference; Importance Sampling; Quadrature methods; Computational algorithms.

3.1. Introduction

The general framework called Layered Adaptive Importance Sampling (LAIS) is a combination of the desirable exploratory behavior of Markov chain Monte Carlo (MCMC) algorithms, and the robustness (and easier theoretical validation) of the importance sampling (IS) schemes [20]. Let us denote with $\bar{\pi}(\mathbf{x}) = \bar{\pi}(\mathbf{x}|\mathbf{y})$ the posterior density in a Bayesian inference problem. The main underlying idea of this algorithm is the *layered* (i.e., hierarchical) procedure for generating samples. In order to generate one sample, a location parameter is drawn from a probability density function (pdf) $\boldsymbol{\mu}_i \sim p(\boldsymbol{\mu})$ (that plays the role of a prior pdf over a location parameter in the hierarchical procedure) and, conditionally on it, a sample is generated from a proposal density centered at $\boldsymbol{\mu}_i$, i.e., $\mathbf{x}_i \sim q_i(\mathbf{x}|\boldsymbol{\mu}_i)$. Then, the sample $\mathbf{x}_i$ is properly weighted according to a *multiple IS* (MIS)

procedure [10, 34]. Hence, the *upper layer* is formed by the generation of $\boldsymbol{\mu}$'s, while in the *lower layer*, we have the generation of $\mathbf{x}$'s and its weighting. More generally, parallel MCMC algorithms addressing different $p_i(\boldsymbol{\mu})$'s, for $i = 1, ..., N$, can be employed to obtain the location parameters $\boldsymbol{\mu}_i$. The use of parallel MCMC chains in the upper layer makes the LAIS framework particularly suitable in multimodal scenarios. Note that the samples $\boldsymbol{\mu}_i$ are not included in the final estimators (just the samples $\mathbf{x}_i$), but only used as location parameters for the proposal densities. In [20], the specific choice $p_i(\boldsymbol{\mu}) = \bar{\pi}(\boldsymbol{\mu})$ has been suggested and successfully tested.

With respect to other benchmark AIS techniques in the literature (see, e.g.,[4]), the LAIS scheme provides very competitive results and exhibits a relevant robustness with respect to tuning of the parameters of proposal densities q_i (such as the scale parameters). The interested reader can observe these properties in the numerical comparison, provided in Section 3.9.1. Moreover, LAIS can be interpreted as:

- An efficient procedure of combining the outputs of several parallel MCMC chains. Several other attempts can be found in the literature (see, e.g., [5]).
- An efficient procedure for estimating the marginal likelihood by using MCMC chains, which is a well-known difficult task for the MCMC techniques [17].

These strengths of the LAIS scheme are very appealing for practitioners and researchers. At the same time, the generic LAIS framework offers a remarkable flexibility which have not been completely exploited in [20], and have been not explored in the further works. For instance, in the upper layer, the user must specify the choices of $p_i(\boldsymbol{\mu})$ and the type of MCMC algorithms; in the lower layer, a specific MIS weighting scheme must be selected. This flexibility allows the LAIS algorithm to handle efficiently different complex inference scenarios, not only multimodality. Introducing specific LAIS schemes for tackling other difficult scenarios of inference is the first main goal of this work. The second main objective of this paper is to describe different procedures for reducing the computational cost of the LAIS scheme.

In this work, as disclosed above, we introduce different schemes for improving the overall performance and reduce the total computational cost. Specifically, we discuss suitable configurations of the LAIS algorithm for addressing the problem of sampling concentrated posteriors (due to complex model or great number of data) and posteriors in high dimensional spaces. This is possible by the use of data-tempered posteriors in the upper layer, that we refer to as *partial posteriors* (see Section 3.4), and advanced MCMC schemes such as Hamiltonian MC (HMC) and sophisticated Gibbs-type techniques (see Section 3.5) [13]. We also discuss different strategies for reducing the overall computational cost. For instance, we propose a procedure for recycling the samples in upper layer and use them in the final estimators, in such a way that the sampling step in the lower layer can be avoided. This drastically reduces the number of evaluations of the

posterior. Moreover, in the lower layer, the cost of weighting can be quite high if we have run long MCMC chains in the upper layer. This problem can also be alleviated by using ideas such as compression or alternative weighting schemes, that reduce the cost but maintain the same performance for the final estimators (see, e.g., [8] or [18]). We test the variants in different scenarios with synthetic and real data. A theoretical discussion about the optimal choice of $p(\boldsymbol{\mu})$ is also provided in the Appendix 3.11.1. Several numerical simulations show the benefits of the proposed LAIS techniques in different challenging sampling problems. Table 3.1 summarizes the main contributions (and the novel schemes) and main acronyms employed in this work. Finally, Table 3.2 summarizes the main notation of the work. In Table 3.2, with the acronym MH, we denote the Metropolis-Hastings algorithm [30]. Related Python and Matlab codes are available at https://github.com/FLlorente/LAIS_extensions.

Table 3.1: Summary of the main contributions and the main acronyms of the work.

Contribution/Proposed scheme	Section	Reducing cost	Improving performance
Partial posteriors LAIS (PLAIS)	3.4	✓	✓
Hamiltonian-driven IS (HMC-LAIS)	3.5		✓
Gibbs-driven IS (Gibbs-LAIS)	3.5		✓
Compressed LAIS (CLAIS)	3.6	✓	
Recycling LAIS (RLAIS)	3.7	✓	
Partial posteriors RLAIS (PA-RLAIS)	3.7	✓	✓
Discussion about the computation cost	3.8	related	
Numerical comparisons	3.9	related	related
Theoretical discussion	App. 3.11.1-3.11		related

Table 3.2: Main notation of work.

$\mathbf{x} \in \mathcal{X}_{\mathtt{tot}} \subseteq \mathbb{R}^{D_X}$	vector of parameters to infer	$\bar{\pi}(\mathbf{x}\|\mathbf{y}_{\mathtt{tot}})$	normalized full posterior
$\mathbf{y}_{\mathtt{tot}}$	data	$\pi(\mathbf{x}\|\mathbf{y}_{\mathtt{tot}})$	unnormalized full posterior
$\mathbf{y}_n$	subset of data	$\bar{\pi}_n(\mathbf{x}\|\mathbf{y}_n)$	normalized partial posterior
D_Y	total number of data (in $\mathbf{y}_{\mathtt{tot}}$)	$\pi_n(\mathbf{x}\|\mathbf{y}_n)$	unnormalized partial posterior
K_n	number of data in $\mathbf{y}_n$	$L(\mathbf{y}\|\mathbf{x})$	likelihood function
$q_n(\mathbf{x}\|\boldsymbol{\mu}_n)$	proposal density in the lower layer	$g(\mathbf{x})$	prior density
$\varphi_n(\mathbf{x}\|\boldsymbol{\mu}_n)$	proposal density within MH (in RLAIS)	$Z = p(\mathbf{y}_{\mathtt{tot}})$	marginal likelihood
$\boldsymbol{\mu}_n$	location parameter (e.g., mean)	$\mathbf{I}$	integral of interest
N	number of the MCMC chains	$w = \frac{\pi(\mathbf{x}\|\mathbf{y}_{\mathtt{tot}})}{\boldsymbol{\Phi}(\mathbf{x})}$	importance weight
T	length of the MCMC chains	$\boldsymbol{\Phi}(\mathbf{x})$	denominator in MIS weights
M	number of samples per proposal	$\boldsymbol{\Psi}(\mathbf{x})$	denominator in MIS weights (in RLAIS)
B	number of sub-regions $\mathcal{X}_m$	$\mathcal{X}_m$	m-th sub-region, $\mathcal{X}_1 \cup \ldots \cup \mathcal{X}_B = \mathcal{X}_{\mathtt{tot}}$

3.2. Problem statement

In many applications, the interest lies in making inference about the vector $\mathbf{x} = [x_1, \ldots, x_{D_X}] \in \mathcal{X}_{\mathtt{tot}} \subseteq \mathbb{R}^{D_X}$. A set of D_Y measurements, $\mathbf{y}_{\mathtt{tot}} = [y_1, y_2, \ldots, y_{D_Y}]$, is received, related to

the variable of interest $\mathbf{x}$. The complete likelihood function is denoted as $L(\mathbf{y}_{\texttt{tot}}|\mathbf{x})$. Considering a prior probability density function (pdf) $g(\mathbf{x})$, the *complete posterior* pdf can be written as

$$\bar{\pi}(\mathbf{x}|\mathbf{y}_{\texttt{tot}}) = \frac{1}{p(\mathbf{y}_{\texttt{tot}})} L(\mathbf{y}_{\texttt{tot}}|\mathbf{x}) g(\mathbf{x}) = \frac{1}{Z} \pi(\mathbf{x}|\mathbf{y}_{\texttt{tot}}), \tag{3.1}$$

where we have denoted $Z = p(\mathbf{y}_{\texttt{tot}})$, and $\pi(\mathbf{x}|\mathbf{y}_{\texttt{tot}}) = L(\mathbf{y}_{\texttt{tot}}|\mathbf{x})g(\mathbf{x})$. Note that $\bar{\pi}(\mathbf{x}|\mathbf{y}_{\texttt{tot}}) \propto \pi(\mathbf{x}|\mathbf{y}_{\texttt{tot}})$ for fixed $\mathbf{y}_{\texttt{tot}}$.

Goal. The objective is to make inference about the variable $\mathbf{x}$ given the information provided by the knowledge of $\mathbf{y}_{\texttt{tot}}$. Generally, this task requires computing integrals of type

$$\mathbf{I} = \int_{\mathcal{X}_{\texttt{tot}}} \mathbf{f}(\mathbf{x}) \bar{\pi}(\mathbf{x}|\mathbf{y}_{\texttt{tot}}) d\mathbf{x}, \tag{3.2}$$

where $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{D_X} \to \mathbb{R}^s$ and $\mathbf{I} \in \mathbb{R}^s$ with $s \geq 1$. When $\mathbf{f}(\mathbf{x}) = \mathbf{x}$, the integral $\mathbf{I}$ represents the *minimum mean square error* (MMSE) estimator of $\mathbf{x}$ [30]. Moreover, we are also interested in the so-called *marginal likelihood*,

$$Z = p(\mathbf{y}_{\texttt{tot}}) = \int_{\mathcal{X}_{\texttt{tot}}} \pi(\mathbf{x}|\mathbf{y}_{\texttt{tot}}) d\mathbf{x}. \tag{3.3}$$

This quantity is particularly useful for the model selection purposes [17, 30]. Generally, we are not able to calculate analytically the integrals above. Importance sampling (IS) and Markov chain Monte Carlo (MCMC) are popular Monte Carlo techniques for approximating integrals as in Eq. (3.2) using random samples [16]. IS provides also an estimator of Eq. (3.3), something that is not straightforward with MCMC (see e.g. [17] for a review of methods for estimating Z). In this work, we consider the LAIS framework which mixes the benefits of MCMC and IS algorithms [20]. In the rest of the work, the dependence on the data $\mathbf{y}$ is often not (explicitly) included in the notation, using for instance $\bar{\pi}(\mathbf{x})$ and $\pi(\mathbf{x})$ instead of $\bar{\pi}(\mathbf{x}|\mathbf{y}_{\texttt{tot}})$ and $\pi(\mathbf{x}|\mathbf{y}_{\texttt{tot}})$.

3.3. Layered adaptive importance sampling (LAIS)

LAIS is an adaptive IS framework that consists of two sampling layers, which are detailed in Table 3.3 and described next. Let $\{q_{n,0}(\mathbf{x}|\boldsymbol{\mu}_{n,0})\}_{n=1}^N$ denote an initial set of N parametric proposals. In the upper layer, the location parameters of the proposals are updated by means of MCMC algorithms. In the simplest case, at iteration t, each $\boldsymbol{\mu}_{n,t-1}$ independently evolves to $\boldsymbol{\mu}_{n,t}$ ($n = 1, \ldots, N$) by running one iteration of a MCMC algorithm with invariant density $p_n(\boldsymbol{\mu})$. More generally, the whole population $\{\boldsymbol{\mu}_{n,t-1}\}_{n=1}^N$ can be updated to $\{\boldsymbol{\mu}_{n,t}\}_{n=1}^N$, e.g., considering more sophisticated population MCMC algorithms [15]. Then, after performing T such iterations, a population of NT location parameters is obtained $\{\boldsymbol{\mu}_{n,t}\}_{n=1}^N$ for all t. In the lower layer, we sample $\mathbf{x}_{n,t} \sim q_{n,t}(\mathbf{x}|\boldsymbol{\mu}_{n,t})$ for $n = 1, \ldots, N$ and

$t = 1, \ldots, T$, and assign weights to each sample.
The weighting procedure is done according to the so-called *deterministic mixture* approach [10]. Some possible choices of the denominator of the importance weights are given in Table 3.4. Clearly, in the specific case of a unique chain $N = 1$, the spatial denominator becomes the standard IS denominator. If N single MCMC steps are performed, i.e., $T = 1$, then the temporal denominator becomes the standard IS denominator. Note that, in LAIS, the adaptation (*upper layer*) is independent from the sampling and weighting steps (*lower layer*). As an example, we can run first, e.g., N parallel chains for T iterations each in order to obtain the NT locations parameters $\{\boldsymbol{\mu}_{n,t}\}$, and then perform standard IS with the NT proposals.
The estimators of Eq. (3.2) and Eq. (3.3) are then given by

$$\widehat{\mathbf{I}} = \frac{1}{NT\widehat{Z}} \sum_{n=1}^{N} \sum_{t=1}^{T} w_{n,t} \mathbf{f}(\mathbf{x}_{n,t}), \tag{3.4}$$

$$\widehat{Z} = \frac{1}{NT} \sum_{t=1}^{T} \sum_{n=1}^{N} w_{n,t}. \tag{3.5}$$

Some bounds and theoretical results related to these estimators can be found in [1].

Table 3.3: LAIS algorithm

Choose $\{q_{n,0}\}_{n=1}^{N}$, $\{\boldsymbol{\mu}_{n,0}\}_{n=1}^{N}$ and the MCMC algorithms in the upper layer.

Upper layer (MCMC).

- **Adaptation:** Apply MCMC transitions with invariant pdf $p_n(\boldsymbol{\mu})$, e.g., $p_n(\boldsymbol{\mu}) = \bar{\pi}(\boldsymbol{\mu}|\mathbf{y}_{\mathtt{tot}})$, i.e.,

$$\{\boldsymbol{\mu}_{n,t-1}\}_{n=1}^{N} \xrightarrow{\text{MCMC}} \{\boldsymbol{\mu}_{n,t}\}_{n=1}^{N}, \quad \forall t = 1, ..., T.$$

Lower layer (IS).

- **Sampling:** $\mathbf{x}_{n,t} \sim q_{n,t}(\mathbf{x}|\boldsymbol{\mu}_{n,t})$, for all n, t.
- **Weighting:**

$$w_{n,t} = \frac{\pi(\mathbf{x}_{n,t}|\mathbf{y}_{\mathtt{tot}})}{\boldsymbol{\Phi}(\mathbf{x}_{n,t})}, \qquad \forall n, t, \tag{3.6}$$

  where different denominators, $\boldsymbol{\Phi}(\mathbf{x}_{n,t})$, are possible. See Table 3.4.

Consistency. The LAIS scheme can be interpreted as a standard, static IS scheme with NT proposals, and the consistency only depends on the proper choice of the denominator $\boldsymbol{\Phi}(\mathbf{x})$ in the importance weights. In Table 3.4, some proper choices, that ensure consistency, are provided which follows the deterministic mixture approach [34]. It is important

Table 3.4: Possible denominators $\boldsymbol{\Phi}(\mathbf{x}_{n,t})$.

complete	**temporal**	**spatial**	**standard**
$\frac{1}{NT}\sum_{\tau=1}^{T}\sum_{i=1}^{N} q_{i,\tau}(\mathbf{x}_{n,t}\|\boldsymbol{\mu}_{i,\tau})$	$\frac{1}{T}\sum_{\tau=1}^{T} q_{n,\tau}(\mathbf{x}_{n,t}\|\boldsymbol{\mu}_{n,\tau})$	$\frac{1}{N}\sum_{i=1}^{N} q_{i,t}(\mathbf{x}_{n,t}\|\boldsymbol{\mu}_{i,t})$	$q_{n,t}(\mathbf{x}_{n,t}\|\boldsymbol{\mu}_{n,t})$

to remark that the consistency does not depend on the choice of the densities $p_n(\boldsymbol{\mu})$ in the upper layer, but, clearly, the efficiency of LAIS is affected by the selected pdfs $p_n(\boldsymbol{\mu})$.

Remark. For the sake of simplicity, we have assumed to draw only one sample $\mathbf{x}_{n,t}$ from each proposal $q_{n,t}(\mathbf{x}|\boldsymbol{\mu}_{n,t})$, in the lower layer. More generally, one could draw $M > 1$ samples, $\mathbf{x}_{n,t}^{(1)}, ..., \mathbf{x}_{n,t}^{(M)}$ from each $q_{n,t}(\mathbf{x}|\boldsymbol{\mu}_{n,t})$. This is often necessary for performing a fair comparison with other AIS techniques and is an additional degree of freedom offered by the LAIS framework (see Sections 3.9.1 and 3.9.4). For simplicity, in the rest of work we consider $M = 1$, unless state otherwise that $M > 1$.

Evaluations of the posterior. In the standard LAIS implementation (i.e. setting $p_n(\boldsymbol{\mu}) = \bar{\pi}(\boldsymbol{\mu}|\mathbf{y}_{\mathtt{tot}})$ for all n), the total number of evaluations E of the posterior is $E = 2NT$ (or, more generally, $E = NT + MNT$), where NT evaluations are performed in the upper layer and NT (or, more generally, MNT) in the lower layer. However, the final estimators only involve $S = NT$ samples. With $M > 1$, the final estimators would involve $S = MNT$ samples.

3.3.1. About the choice of the denominator

The computation of the weights in the lower layer allows for different possible denominators, shown in Table 3.4. The function $\boldsymbol{\Phi}(\mathbf{x}_{n,t})$ can be taken to be the proposal that actually generated $\mathbf{x}_{n,t}$ (*standard*), the mixture of proposals across different chains (*spatial*), the mixture of proposals within the chain (*temporal*), or the mixture of all proposals (*complete*). Note that, we always have the evaluation of the complete posterior in the numerator, hence all the weighting strategies have the same number of posterior evaluations, i.e., NT. However, in practice, the cost of the *complete*, *temporal* and *spatial* weighting schemes is higher than the *standard* one, and it will increase the overall computation time. This is more obvious in real applications where many chains are run for a long time, i.e., T and N are very large. Commonly, $T \gg N$, so that the *spatial* scheme is cheaper than the *temporal* scheme, and both are much cheaper than the *complete* scheme. In return, these schemes can produce a remarkable improvement in the performance of the final estimators. It can be theoretically proved that the deterministic mixture denominators produce estimators with lower (or equal) variance than the standard weighting [10]. Indeed, our experiments in Section 3.9.3 show that the complete denominator consistently produces more stable estimators with only a small increase in computational cost, as compared to

the overall cost of the algorithm.

3.3.2. Elements for the design of a specific LAIS implementation

A specific implementation of the LAIS algorithm is determined by the choices of

1. the invariant densities $p_n(\boldsymbol{\mu})$;
2. the MCMC approach (e.g., parallel or single longer chain Metropolis-Hastings, advanced MCMC schemes, etc.);
3. the proposals $q_{n,t}(\mathbf{x}|\boldsymbol{\mu}_{n,t})$; and
4. the denominator $\boldsymbol{\Phi}(\mathbf{x})$.

Namely, a particular LAIS implementation is completely defined by the choice of those four elements. Below, we present several variants and improvements for the LAIS framework concerning each one of the elements above. For instance, regarding the pdfs $p_n(\boldsymbol{\mu})$, we describe the suitable use of different type of tempered posteriors. The application of sophisticated MCMC algorithms in the upper layer is also discussed. Recycling sample schemes (which involve the selection of proposals $q_{n,t}$ as well) and the design of cheap denominators $\boldsymbol{\Phi}$ in the lower layer are also introduced in the next sections.

3.4. Data tempering and partial posteriors in the upper layer

The LAIS framework has flexibility in the upper layer design of selecting different invariant densities $p_n(\boldsymbol{\mu})$. A theoretical discussion regarding the optimal choice of the invariant densities, $p_n(\boldsymbol{\mu})$, is given in Appendix 3.11.1. In this section, we introduce the possibility of using *partial posteriors* (i.e., posteriors considering a reduced number of data) as invariant pdfs $p_n(\boldsymbol{\mu})$. The benefit is twofold: (a) reducing the cost of the posterior evaluations in the upper layer, and (b) helping the space exploration of MCMC chains. This second effect is often called *data tempering*. See Appendix 3.11.1 for further details.
Specifically, let $\mathbf{y}_n \in \mathbb{R}^{K_n}$ denote a subset of data points, i.e., $\mathbf{y}_n \subset \mathbf{y}_{\mathtt{tot}}$ (with $K_n \ll D_Y$) and assume we have N subsets $\mathbf{y}_1, \ldots, \mathbf{y}_N$. For the sake of simplicity, we assume that $\{\mathbf{y}_n\}_{n=1}^N$ represents a partition of $\mathbf{y}_{\mathtt{tot}}$, i.e., N non-overlapping pieces such that $\sum_{n=1}^N K_n = D_Y$. However, more generally, we could also have $\mathbf{y}_n \cap \mathbf{y}_{n'} \neq \emptyset$. Note that we are keeping the vector notation for data subset $\mathbf{y}_n$ but sometimes we use it as a set notation, just for the sake of simplicity. Hence, let us define the partial posteriors, for using them as invariant densities in the upper layer,

$$p_n(\mathbf{x}) = \bar{\pi}_n(\mathbf{x}|\mathbf{y}_n) \propto L_n(\mathbf{y}_n|\mathbf{x})g_n(\mathbf{x}), \tag{3.7}$$

where $L_n(\mathbf{y}_n|\mathbf{x})$ is the likelihood of the batch $\mathbf{y}_n$, and $g_n(\mathbf{x})$ plays the role of a *partial* prior pdf. For our purpose, we can keep $g_n(\mathbf{x}) = g(\mathbf{x})$ for all n, or we can split the prior

contribution into each data subset, for instance, setting $g_n(\mathbf{x}) = g(\mathbf{x})^{\frac{1}{N}}$ for all n, which is a typical choice in several distributed settings (motivated so that the product of $\bar{\pi}_n(\mathbf{x}|\mathbf{y}_n)$ is proportional to the complete posterior) (e.g., see [33]) . Therefore, the partial posterior $\bar{\pi}_n(\mathbf{x}|\mathbf{y}_n)$ is a tempered version of the complete posterior since its likelihood $L_n(\mathbf{y}_n|\mathbf{x})$ is less informative, i.e., wider, than in the case where we consider all data.
Thus, we consider that each MCMC chain in the upper layer addresses a different partial posterior $p_n(\mathbf{x}) = \bar{\pi}_n(\mathbf{x}|\mathbf{y}_n)$ $(n = 1, \ldots, N)$. Hence, there are as many chains as number of partial posteriors. We call this scheme as *partial posteriors LAIS* (PLAIS) method. Note that, in PLAIS, we still evaluate the complete posterior in the lower layer, so the total number of full posterior evaluations is NT (in the lower layer). Furthermore, the use of partial posteriors produces more dispersed location parameters of the proposals in the lower layer. This increases the robustness of the method, since it reduces the chance of obtaining huge weight values and, as a consequence, avoids IS estimators with infinite variance (see the example 1 in [17]).

3.5. Hamiltonian and Gibbs-driven importance samplers

The simplest choice of MCMC schemes in the upper layer is a unique Metropolis-Hastings (MH) chain, or to employ N independent parallel MH algorithms. However, more sophisticated algorithms can be considered (such as Langevin, Hamiltonian and Gibbs samplers), which can further enhance the performance of the algorithm. On the other hand, the LAIS algorithm can be interpreted as a way to help these MCMC schemes to improve their efficiency and allow them to estimate efficiently the marginal likelihood Z (as shown in the numerical experiments in Section 3.9).

Hamiltonian MC in the upper layer. The Hamiltonian Monte Carlo (HMC) algorithm is usually considered as the state-of-the-art technique in the MCMC world [25]. However, as with the rest of MCMC methods, it is not straightforward to estimate the marginal likelihood with HMC samples [17]. Additionally, it is well-known the difficulty of tuning its hyperparameters for obtaining efficient sampling [25]. In this context, we propose using different HMC algorithms in the upper layer in Table 3.3 , each chain employing possibly different parameters. Thus, several sets of parameters are jointly used. Note also that we do not need to fine-tune the hyperparameters since the states in the upper layer are not used directly as samples in our framework. The lower layer in the LAIS scheme provides a straightforward estimation of the marginal likelihood. We compare the performance of these algorithms, denoted as HMC-LAIS, with HMC in Sect. 3.9.3.

Gibbs algorithms in the upper layer. Another possibility is to use Gibbs samplers in the upper layer [30]. The Gibbs sampler is component-wise scheme, i.e., at each iteration each component of the parameter vector $\mathbf{x}$ is drawn from the corresponding full-conditional density keeping fixed the rest of components. Thus, they have the advantage

of working in lower dimension at each iteration, which allows the design more efficient samplers in high dimensional spaces. For instance, extremely efficient MH-within-Gibbs algorithms can be designed using Adaptive Rejection Metropolis schemes for drawing from each one-dimensional full-conditional (e.g., see [11] and [23]). Another important benefit is the use of a Gibbs sampler is particularly useful for drawing from very tight posteriors, as shown in [22] (see also Section 3.9.5).

More generally, the joint use of HMC, Langevin, and Gibbs-based schemes can be potentially applied in the upper layer. For instance, an extension of the Gibbs sampling idea is the so-called adaptive direction sampling, which can speed up the mixing of generated chains, choosing different one-dimensional direction of sampling at each iteration [12]. Note that both, HMC-LAIS and Gibbs-LAIS, are very useful schemes for sampling from concentrated/tight posteriors or high-dimensional posteriors (see the numerical simulations in Section 3.9).

3.5.1. Optimizers versus samplers

Let us consider for simplicity the choice $p_n(\boldsymbol{\mu}) = \bar{\pi}(\boldsymbol{\mu})$ suggested in [20]. Instead of sampling, a simpler alternative could be simply to perform optimization steps for obtaining the location parameters $\boldsymbol{\mu}_i$. However, a sampler takes into account not just the modes of $\bar{\pi}(\boldsymbol{\mu})$ but all the probability mass around these modes. Therefore, using a sampler, location parameters $\bar{\pi}(\boldsymbol{\mu})$ would be spread out in the regions of high probability mass (not only at the modes; or close to the modes). This aspect ensures and induces robustness in the IS scheme which uses proposal densities with location parameters $\boldsymbol{\mu}_i$, since the full-mixture of proposal densities tends to have a greater variance than the variance of posterior distribution. See Appendix 3.11.1, for further details. This property is extremely important since it avoids the catastrophic scenario of infinite variance in the final IS estimators, which can occur when the proposal density has smaller variance than the target pdf (see the illustrative example 1 in [17]).

3.5.2. Upper layer design: a summary

So far (in Sections 3.4 and 3.5), we have proposed strategies for improving the efficiency of the final estimators of LAIS, focusing so far on the upper layer in Table 3.3. These enhancements are particularly relevant in different challenging inference scenarios, such as tight posteriors and/or high dimensional problems. For other complex settings, such as multimodal posteriors, the use of parallel MCMC chains (already suggested in [20]) is important. Table 3.5 outlines the correspondence between inference scenarios (as well as other features and benefits) and the proposed procedures to employ in the upper layer. For instance, the data tempering is useful in multimodal and high-dimensional scenarios, and particularly useful in the case of concentrated posteriors. The cost of running the upper layer gets also reduced when using partial posteriors since the MCMC algorithms

do not require to process all the data at each iteration. Moreover, the data tempering generally increases the robustness of the LAIS algorithm. Last but not least, observe that all the techniques can be employed jointly in the upper layer, for instance, parallel HMC (or Gibbs) chains (with different parameters) considering each one a different partial posterior. In this sense, LAIS can ensure good and robust performance. See Section 3.9 for further details.

Table 3.5: Table of correspondence between benefits and inference scenarios, versus the proposed procedures (and methods) in the upper layer (✓ = useful, and ★ = very useful).

Methods/ Procedures (upper layer)	**Multimodality/ helping the exploration**	**Robustness (e.g., to the choice of proposal parameters)**	**concentrated/tight posteriors**	**high dimensional spaces**
parallel chains	★	★		
data-tempering	✓	✓	★	✓
HMC-driven			✓	★
Gibbs-driven			★	✓

3.6. Compression for parsimonious sampling and weighting

The *complete* weighting scheme (see Table 3.4) provides the best performance in terms of variance, at the expense of an increase in the computational cost, especially in real applications since T and N can be very large (note that it requires NT proposal evaluations per sample). One possibility in order to reduce this cost, without decreasing T or N, is the use of partial MIS denominators [10]. Another approach consists in using some technique that summarizes the population of NT samples. A first attempt has been provided in [8]. Another possible way is to apply a compression of Monte Carlo samples [18], as we describe below. These schemes reduce the cost of both sampling and weighting in the lower layer.

Compressed LAIS (CLAIS). Let consider a set of R means $\{\boldsymbol{\mu}_k\}_{k=1}^R$ generated by MCMC in the upper layer, and let B be a constant value such that $B < R$. Note that, in the case of N parallel chains of length T in the upper layer, we have $R = NT$. Given a partition of $\mathcal{X}_{\texttt{tot}}$, i.e., $\mathcal{X}_1 \cup \mathcal{X}_2 \cup \ldots \cup \mathcal{X}_B = \mathcal{X}_{\texttt{tot}}$ formed by convex, disjoint sub-regions $\mathcal{X}_m$, we denote the subset of the set of indices $\{1, \ldots, R\}$,

$$\mathcal{J}_m = \{i = 1, \ldots, R : \boldsymbol{\mu}_i \in \mathcal{X}_m\}, \qquad m = 1, ..., B,$$

which are associated with the samples in the m-th sub-region $\mathcal{X}_m$. The partition $\mathcal{X}_1 \cup \mathcal{X}_2 \cup \ldots \cup \mathcal{X}_B = \mathcal{X}_{\texttt{tot}}$ can be obtained using some a-priori information or, as an example, by means of a clustering method. The cardinality $|\mathcal{J}_m|$ denotes the number of samples in $\mathcal{X}_m$ and we have $\sum_{m=1}^B |\mathcal{J}_m| = R$. We can compress the information contained in samples, constructing a stratified approximation based on B weighted particles $\{\mathbf{s}_m, a_m\}_{m-1}^B$, where

$\mathbf{s}_m$ is a (properly chosen) point in $\mathcal{X}_m$ and $a_m = \frac{|J_m|}{R}$.

Possible choices of $\mathbf{s}_m$. The summary points $\mathbf{s}_m$ can be randomly chosen, picking uniformly a mean in $\mathcal{X}_m$, in the set $\{\boldsymbol{\mu}_i\}_{i\in\mathcal{J}_m}$ or using a deterministic procedure, e.g.,

$$\mathbf{s}_m = \frac{1}{|J_m|}\sum_{j\in J_m}\boldsymbol{\mu}_j. \tag{3.8}$$

For the statistical properties of these choices see [18]. Other choices based on empirical quantiles are also possible. As an example, a suitable compression scheme can be provided applying a clustering method to the set $\{\boldsymbol{\mu}_k\}_{k=1}^R$, where B represents the number of clusters. After the compression, we can consider as proposal and denominator in the lower layer the following mixture of densities $p(\mathbf{x}|\mathbf{s},\boldsymbol{\Sigma})$ where $\mathbf{s}$, $\boldsymbol{\Sigma}$ represent a location parameter and a covariance matrix,

$$q_B(\mathbf{x}) = \sum_{m=1}^{B} a_m p\left(\mathbf{x}|\mathbf{s}_m,\boldsymbol{\Sigma}\right). \tag{3.9}$$

Thus, the mixture q_B is used for sampling and computing the weights in the lower layer. A suitable choice of $\mathbf{s}_m$ and $\boldsymbol{\Sigma}$ is the key point for the success of the compressed scheme. For the summary points $\mathbf{s}_m$, we suggest the use of the deterministic procedure in Eq. (3.8).

Suitable choice of $\boldsymbol{\Sigma}$. We suggest to obtain the $D_X \times D_X$ covariance matrix $\boldsymbol{\Sigma}$ as

$$\boldsymbol{\Sigma} \;=\; \mathbf{Q}_\mu - \mathbf{Q}_C + \sigma_p^2\mathbf{I}. \tag{3.10}$$

where $\mathbf{Q}_\mu = \frac{1}{R}\sum_{k=1}^R (\boldsymbol{\mu}_k - \mathbf{m})(\boldsymbol{\mu}_k - \mathbf{m})^\top$ with $\mathbf{m} = \frac{1}{R}\sum_{k=1}^R \boldsymbol{\mu}_k$ is the covariance matrix of all R means $\boldsymbol{\mu}_k$, and $\mathbf{Q}_C = \sum_{m=1}^B a_m (\mathbf{s}_m - \mathbf{m}_C)(\mathbf{s}_m - \mathbf{m}_C)^\top$ with $\mathbf{m}_C = \sum_{m=1}^B a_m \mathbf{s}_m$ is the covariance matrix of the summary samples. Clearly, if $\mathbf{s}_m$ are chosen as in Eq. (3.8), then $\mathbf{m} = \mathbf{m}_C$. Finally, σ_p^2 is chosen by the user. With $\mathbf{s}_m$ in Eq. (3.8), it is possible to show that

$$\mathbf{Q}_\mu - \mathbf{Q}_C = \sum_{m=1}^{B} a_m \left(\frac{1}{|J_m|}\sum_{j\in\mathcal{J}_m}\left(\boldsymbol{\mu}_j - \mathbf{s}_m\right)\left(\boldsymbol{\mu}_j - \mathbf{s}_m\right)^\top\right). \tag{3.11}$$

That is, the covariance of each component in $q_B(\mathbf{x})$ is the weighted average of the covariances within clusters plus the term $\sigma_p^2\mathbf{I}$. We remark that a suitable choice of $\boldsymbol{\Sigma}$ is crucial for the performance of the compression technique. The proposed covariance matrix $\boldsymbol{\Sigma}$ in Eq. (3.10) is a robust choice which provides good performance, as shown in Section 3.9.3, and below we explain the reasons.

The combined choice of $\mathbf{s}_m$ in Eq. (3.8) and $\boldsymbol{\Sigma}$ in (3.10) has the following property. Let us assume the use of denominator with B components ($1 \le B \le R$) in the mixture $q_B(\mathbf{x})$. Without compression, we have $B = R$, $\mathbf{s}_k = \boldsymbol{\mu}_k$, $\mathbf{Q}_\mu = \mathbf{Q}_C$, so we have the covariance of each mixture component is $\boldsymbol{\Sigma} = \sigma_p^2\mathbf{I}$, as expected. With the maximum compression, $B = 1$, then $\mathbf{Q}_C$ is the null matrix and $\boldsymbol{\Sigma} = \mathbf{Q}_\mu + \sigma_p^2\mathbf{I}$. Hence, with maximum compression, the proposal q_B takes into account the dispersion set by the user (by the term $\sigma_p^2\mathbf{I}$) plus

the covariance matrix of the R location parameters $\boldsymbol{\mu}_k$ (i.e., the term $\mathbf{Q}_\mu$), obtained in the upper layer. Finally, note that the cost of the employed compression technique must be lower than the cost of evaluating the full denominator. We test the performance of CLAIS with several choices of R, and compare it with standard LAIS in Section 3.9.3.

3.7. Recycling LAIS (RLAIS)

In this Section, we discuss the possibility of recycling the samples, and their corresponding evaluations, from the upper layer for their use in the lower layer, hence reducing the overall computational cost. For simplicity, let us assume the use of N parallel Metropolis-Hastings (MH) algorithms in the upper layer. Moreover, in this first part of the section, assume that $p_n = \bar{\pi}$ for all n. Given the initial state $\boldsymbol{\mu}_{n,0}$, a proposal pdf φ_n, and a length value T, the n-th MH chain follows the following steps:

- **For** $t = 1, \dots, T$**:**

  1. Draw $\mathbf{z}_{n,t} \sim \varphi_n(\mathbf{x}|\boldsymbol{\mu}_{n,t-1})$.

  2. Set $\boldsymbol{\mu}_{n,t} = \mathbf{z}_{n,t}$ with probability

$$\alpha = \min\left[1, \frac{\pi(\mathbf{z}_{n,t}|\mathbf{y}_{\mathtt{tot}})\varphi_n(\boldsymbol{\mu}_{n,t-1}|\mathbf{z}_{n,t})}{\pi(\boldsymbol{\mu}_{n,t-1}|\mathbf{y}_{\mathtt{tot}})\varphi_n(\mathbf{z}_{n,t}|\boldsymbol{\mu}_{n,t-1})}\right], \tag{3.12}$$

     otherwise, set $\boldsymbol{\mu}_{n,t} = \boldsymbol{\mu}_{n,t-1}$ (with probability $1 - \alpha$).

- **Outputs:** The chain $\{\boldsymbol{\mu}_{n,t}\}_{t=0}^{T-1}$. Additionally, we obtain and store $\{\mathbf{z}_{n,t}\}_{t=1}^{T}$, $\{\pi(\mathbf{z}_{n,t}|\mathbf{y}_{\mathtt{tot}})\}_{t=1}^{T}$ and $\{\varphi_n(\mathbf{z}_{n,t}|\boldsymbol{\mu}_{n,t-1})\}_{t=1}^{T}$.

Therefore, at each iteration, a candidate is drawn $\mathbf{z}_{n,t} \sim \varphi_n(\mathbf{x}|\boldsymbol{\mu}_{n,t-1})$ and then it is tested (accepted or discarded) as possible new state, according to the acceptance MH probability. If we store all candidates $\{\mathbf{z}_{n,t}\}_{t=1}^{T}$ and the corresponding evaluations of the posterior $\{\pi(\mathbf{z}_{n,t}|\mathbf{y}_{\mathtt{tot}})\}_{t=1}^{T}$ (for all n), required in the computation of α in Eq. (3.12), we can use them in the lower layer as samples, i.e., we set $\mathbf{x}_{n,t-1} = \mathbf{z}_{n,t}$. In this way, we reduce the computation time since we do not need to draw additional samples.
Note that $\varphi_n(\mathbf{x}|\boldsymbol{\mu}_{n,t-1})$ becomes the proposal in the lower layer, i.e., we set $q_{n,t}(\mathbf{x}) = \varphi_n(\mathbf{x}|\boldsymbol{\mu}_{n,t-1})$. The evaluations of the proposal $\varphi_n(\mathbf{z}_{n,t}|\boldsymbol{\mu}_{n,t-1})$ can be also stored. Depending on the choice of the weighting scheme, other evaluations of different proposals φ_j, with $j \neq n$, can be required. This also produces a slight reduction of the cost of evaluating the denominator of the weights in the lower layer. See the next section for further details. The algorithm is outlined in Table 3.6, and Table 3.7 shows different weighting procedures. Since $p_n = \bar{\pi}$ and the posterior evaluations are recycled, the total number of posterior evaluations in RLAIS is only $E = NT$.

Consistency. It is important to note that we can find an equivalent proposal $\widetilde{q}_{MH}(\mathbf{x})$ of

MH-type algorithms which can be expressed as a convolution integral, similarly as we have done in LAIS. See the Appendix 3.11 for more details. In RLAIS, the different MIS denominators can be considered as Monte Carlo approximations of this equivalent proposal $\widetilde{q}_{MH}$, expressed as an integral in Eq. (3.24). Therefore, in the case of the first 3 different MIS denominators (the complete, spatial and temporal mixtures) as N and T grow, the chosen denominator provides a better approximation of the $\widetilde{q}_{MH}$ and the MIS weights becomes closer and closer to standard importance weights of the form $w_{n,t} = \frac{\pi(\mathbf{x}_{n,t}|\mathbf{y}_{\mathtt{tot}})}{\widetilde{q}_{MH}(\mathbf{x}_{n,t})}$. RLAIS can be seen as a multiple-chain generalization of [31, 32].

Table 3.6: LAIS with recycling (RLAIS)

1. **Sampling:** Let consider Metropolis-Hastings (MH)-type schemes with random walk proposal densities $\varphi_{n,t}(\mathbf{x}|\boldsymbol{\mu}_{n,t})$ ($\varphi_{n,t}$ can vary with t since we assume they can be also adaptive schemes), generating N MCMC chains of length T.

   Then, the states of the chains are $\boldsymbol{\mu}_{n,t}$, for $n = 1, \dots, N$ and $t = 1, ..., T$. At each iteration of one MH scheme, we draw a candidate $\mathbf{z}_{n,t} \sim \varphi_{n,t}(\mathbf{x}|\boldsymbol{\mu}_{n,t-1})$ that will be accepted or rejected in the MH step. We save all the NT candidates $\mathbf{z}_{n,t}$ for $n = 1, \dots, N$ and $t = 1, ..., T$.

2. **Weighting:** Assign to $\mathbf{z}_{n,t}$ the weights

$$w_{n,t} = \frac{\pi(\mathbf{z}_{n,t}|\mathbf{y}_{\mathtt{tot}})}{\boldsymbol{\Psi}(\mathbf{z}_{n,t})}, \tag{3.13}$$

   where different possible choices for $\boldsymbol{\Psi}(\mathbf{z}_{n,t})$ are possible (see Table 3.7).

3. **Output:** Return all the pairs $\{\mathbf{z}_{n,t}, w_{n,t}\}$, and/or the estimators given in Eqs (3.5) and (3.4).

Table 3.7: Possible denominators $\boldsymbol{\Psi}(\mathbf{x}_{n,t})$.

complete	**temporal**	**spatial**	**standard**
$\frac{1}{NT}\sum_{\tau=0}^{T-1}\sum_{n=1}^{N}\varphi_{n,\tau}(\mathbf{x}_{n,t}\|\boldsymbol{\mu}_{n,\tau})$	$\frac{1}{T}\sum_{\tau=0}^{T-1}\varphi_{n,\tau}(\mathbf{x}_{n,t}\|\boldsymbol{\mu}_{n,\tau})$	$\frac{1}{N}\sum_{n=1}^{N}\varphi_{n,t}(\mathbf{x}_{n,t}\|\boldsymbol{\mu}_{n,t})$	$\varphi_{n,t}(\mathbf{x}_{n,t}\|\boldsymbol{\mu}_{n,t})$

PLAIS with recycling (PA-RLAIS). We can combine the idea of using the partial posteriors and the RLAIS approach. Indeed, also in PLAIS, it is possible to avoid the sampling step if we recycle all candidates produced within the MH algorithms in the upper layer. We denote the resulting scheme as PA-RLAIS. We can recycle the candidates $\{\mathbf{z}_{n,t}\}_{t=1}^{T}$ and the proposal evaluations $\{\varphi_n(\mathbf{z}_{n,t}|\boldsymbol{\mu}_{n,t-1})\}_{t=1}^{T}$ (for all n) but, in this scenario, we have not evaluations of the full posterior in the upper layer (then we cannot recycle the posterior evaluations).

3.8. Computation costs of the proposed schemes

Generally, the most costly step is the evaluation of the complete posterior $\pi(\mathbf{x}|\mathbf{y}_{\texttt{tot}})$ (due to a costly model or number of data). The evaluation of the partial posteriors is not that costly since we choose the batch sizes such that $K_n \ll D_Y$ for all $n = 1, \ldots, N$. Thus, the comparison among PLAIS, RLAIS and PAPIS, as well as with other methods, must be done in terms of number of evaluations of the (unnormalized) posteriors, the complete posterior $\pi(\mathbf{x})$, and/or the partial posteriors $\pi_n(\mathbf{x})$'s. A summary of the number of evaluations of $\pi(\mathbf{x})$ and all partial posteriors $\pi_n(\mathbf{x})$'s is given below:

Method	**Upper layer**		**Lower layer**	**Drawing samples**
	evals of $\pi(\mathbf{x}\|\mathbf{y}_{\texttt{tot}})$	evals of $\pi(\mathbf{x}\|\mathbf{y}_n)$	evals of $\pi(\mathbf{x}\|\mathbf{y}_{\texttt{tot}})$	**in the lower layer**
LAIS	NT	0	NT	✓
PLAIS	0	NT	NT	✓
RLAIS	NT	0	0	X
PA-RLAIS	0	NT	NT	X
—	—	cheaper	—	—
CLAIS can be also combined with the other schemes above for building cheaper denominators.				

Therefore, the total number of full-posterior evaluations of the standard LAIS scheme is $E = NT + NT = 2NT$. If we draw $M > 1$ samples from each proposal density $q_{n,t}$ in the lower layer, the total number of full-posterior evaluations is $E = NT + MNT = (M+1)NT$.

If we denote as C the *atomic cost* of evaluating once the likelihood function with only one data point, then the total cost associated to the total number of the target evaluations (considering evaluations of full-posterior and/or evaluations of partial posteriors) of the different techniques is given below:

Method	**Total cost associated to the posterior evaluations**
LAIS	$2NTCD_Y$
PLAIS	$TC\left(\sum_{n=1}^{N} K_n\right) + NTCD_Y$ $= TCD_Y + NTCD_Y = (N+1)TCD_Y$
RLAIS	$NTCD_Y$
PA-RLAIS	$TCD_Y + NTCD_Y = (N+1)TCD_Y$

where N is the number of chains (with length T) in the upper layer, D_Y is the total number of data, and C is the atomic cost previously described. We have used that $\sum_{n=1}^{N} K_n = D_Y$ where K_n are the number of data in the n-th partial posterior. Clearly, RLAIS and standard LAIS are the algorithms with lowest and greatest costs, respectively, as shown below.

Inequalities in terms of cost of total posterior evaluations:				
Cost of RLAIS	<	Cost of PA-RLAIS = Cost of PLAIS	<	Cost of LAIS
$\Downarrow$		$\Downarrow$		$\Downarrow$
$NTCD_Y$	<	$(N+1)TCD_Y$ = $(N+1)TCD_Y$	<	$2NTCD_Y$

However, considering also the cost of sampling from the proposal pdfs, PA-RLAIS is less costly than PLAIS since it does not require extra samples in the lower layer. This is an additional advantage of RLAIS as well. We recall that the reason of using partial posteriors is not only a reduction on the computational cost. Indeed, the use of partial posteriors fosters the space exploration due to the data-tempering effect. Finally, we also remark that the overall computational cost also depends on the denominator choice: this is the reason of employing the proposed scheme in Section 3.6, denoted as CLAIS. The number of proposal evaluations per sample *in the lower layer* with the different possible denominators is given below:

Method	**complete**	**temporal**	**spatial**	**standard**
Stand. LAIS	NT	T	N	1
RLAIS	$NT-1$	$T-1$	$N-1$	0

Recall that, for simplicity, throughout this work we have considered to draw $M = 1$ sample from each proposal, in the lower layer. However, all the formulas above just suffer some mild changes for $M > 1$.

3.9. Numerical experiments

In this section, we test the performance of the algorithms described in this work. We have considered different challenging scenarios. As an example, we tackle multimodal target densities (in Sections 3.9.1 and 3.9.4), high-dimensional problems (in Section 3.9.4) and extremely sharp/tight posteriors (in Section 3.9.5). In the last experiment (Section 3.9.6), we also analyze real data in a regression problem on the daily deaths during the COVID-19 pandemic in Italy. The correspondence between proposed algorithms and sections is given below:

Method	**Section 3.9.1**	**Section 3.9.2**	**Section 3.9.3**	**Section 3.9.4**	**Section 3.9.5**	**Section 3.9.6**
Stand. LAIS	✓	✓				
PLAIS		✓				
CLAIS			✓			
RLAIS					✓	
PA-RLAIS		✓				
HMC-LAIS			✓	✓		
Gibbs-LAIS					✓	✓
Diff. Den. $\mathbf{\Phi(x)}$			✓			

3.9.1. Comparison with benchmark AIS schemes

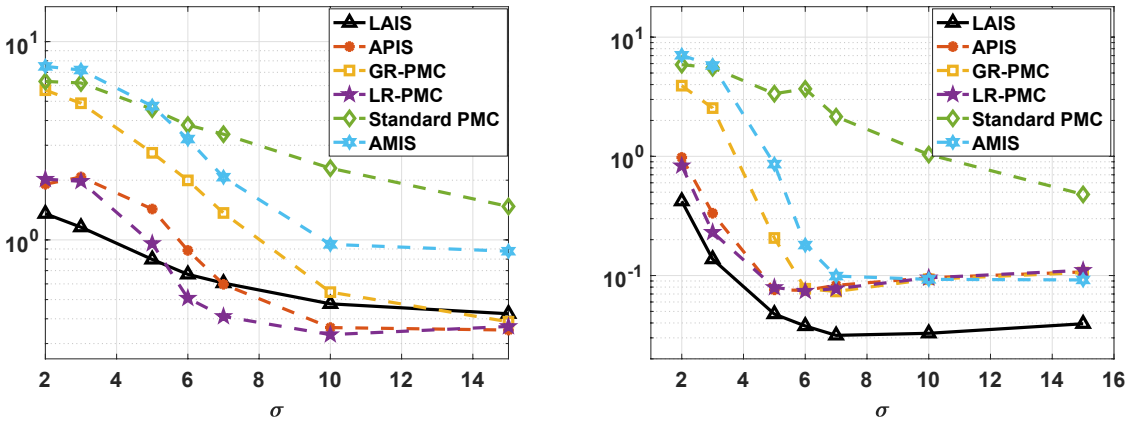
In this section, we compare LAIS with the most relevant and benchmark AIS schemes proposed in the literature [4, 6, 9, 19]. The objective of this section is to highlight the robustness of the LAIS scheme with respect to the choice of the parameters, comparing with the results of the other AIS techniques. With this aim, we consider a highly-multimodal bivariate target pdf defined as a mixture of five Gaussians, i.e.,

$$\pi(\mathbf{x}) = \frac{1}{5}\sum_{i=1}^{5} \mathcal{N}(\mathbf{x}; \boldsymbol{\nu}_i, \boldsymbol{\Lambda}_i), \quad \mathbf{x} \in \mathbb{R}^2, \tag{3.14}$$

where $\mathcal{N}(\mathbf{x}; \boldsymbol{\nu}_i, \boldsymbol{\Lambda}_i)$ denotes a Gaussian density with mean vector $\boldsymbol{\nu}_i$ and covariance matrix $\boldsymbol{\Lambda}_i$, $\nu_1 = [-10, -10]^\top$, $\boldsymbol{\nu}_2 = [0, 16]^\top$, $\boldsymbol{\nu}_3 = [13, 8]^\top$, $\boldsymbol{\nu}_4 = [-9, 7]^\top$, $\boldsymbol{\nu}_5 = [14, -14]^\top$, $\boldsymbol{\Lambda}_1 = [2,\ 0.6; 0.6,\ 1]$, $\boldsymbol{\Lambda}_2 = [2,\ -0.4; -0.4,\ 2]$, $\boldsymbol{\Lambda}_3 = [2,\ 0.8; 0.8,\ 2]$, $\boldsymbol{\Lambda}_4 = [3,\ 0; 0,\ 0.5]$, and finally $\boldsymbol{\Lambda}_5 = [2,\ -0.1; -0.1,\ 2]$. This is a very challenging scenario since we have 5 different modes, far away one from another. In this example, we can analytically compute different moments of the target in (3.14), and therefore we can easily validate the performance of the different techniques. In particular, we consider the computation of the mean of the target, $E[\mathbf{X}] = [1.6, 1.4]^\top$, and the normalizing constant, $Z = 1$, for $\mathbf{X} \sim \frac{1}{Z}\pi(\mathbf{x})$. We compute the mean squared error (MSE) in the estimation of $E[\mathbf{X}]$ and in the normalizing constant Z (which usually represents a marginal likelihood, when the density of interest is a Bayesian posterior).

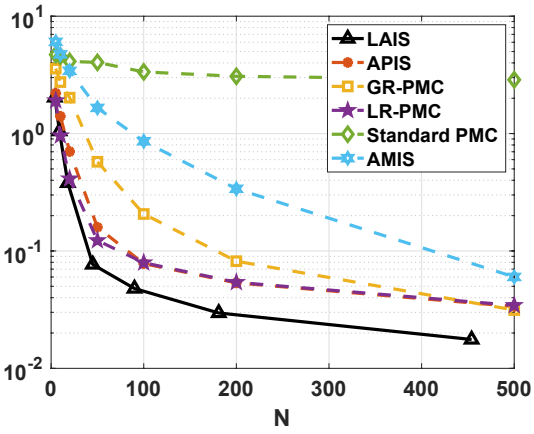
We apply LAIS with N parallel MH chains in the upper layer (of length T). We assume Gaussian proposal densities for all of the methods compared, and deliberately choose a bad initialization of the means in order to test the robustness and the adaptation capabilities. Specifically, the initial location parameters of the proposals are selected uniformly within the $[-4, 4] \times [-4, 4]$ square, i.e., $\boldsymbol{\mu}_{n,0} \sim \mathcal{U}([-4, 4] \times [-4, 4])$ for $n = 1, \ldots, N$. Note that none of the modes of the target are contained within this initialization square. We test all the alternatives using the same isotropic covariance matrices for all the Gaussian proposals, $\mathbf{C}_n = \sigma^2\mathbf{I}_2$, where in some simulations we vary σ. All the results have been averaged over 10^3 independent runs, where the total number of target evaluations E is the same in all the techniques (see Section 3.8 for LAIS). In order to make possible a fair comparison with other schemes, in LAIS we draw $M > 1$ samples from each proposal density $q_{n,t}$ in the lower layer, so that the total number of full-posterior evaluations in LAIS is $E = NT + MNT = (M + 1)NT$ (as shown in the previous section). We apply also the following schemes: the standar **Population Monte Carlo (PMC)** technique [4], the **Adaptive Population Importance Sampling (APIS)** method [19], the improved PMC schemes **GR-PMC** and **LR-PMC** [9], and the **Adaptive Multiple Importance Sampling (AMIS)** approach [6]. We remark that all the comparisons have been performed with the same number of target evaluations E.
For instance, in Figure 3.1(a), we vary the standard deviation of the proposal densities σ, and we set $N = 10$, $M = 9$, $T = 100$ for LAIS, $N = 10$, $M = 10$ $T = 100$ for APIS, GR-PMC and LR-PMC, and $M = 100$ and $T = 100$ in AMIS (since in AMIS we have a unique

proposal density). We repeat the experiment in Figure 3.1(b), but considering $N = 100$. In Figure 3.1(c), we set $\sigma = 5$ and vary N. We can observe that stand. LAIS generally outperforms the other techniques. Even when LAIS does not provide the smallest MSE, it obtains close results. Namely, LAIS provides competitive results for any of the values σ or N, proving its robustness. As N grows, LAIS becomes even more competitive.



(a) $N = 10$

(b) $N = 100$



(c) $\sigma = 5$

Figure 3.1: MSE in log-scale obtained by different techniques, of the experiment in Section 3.9.1. **(a)** With $N = 10$, and varying σ; **(b)** with $N = 100$ and varying σ; **(c)** with $\sigma = 5$ and varying N.

3.9.2. Parameter fitting in a non-linear regression problem

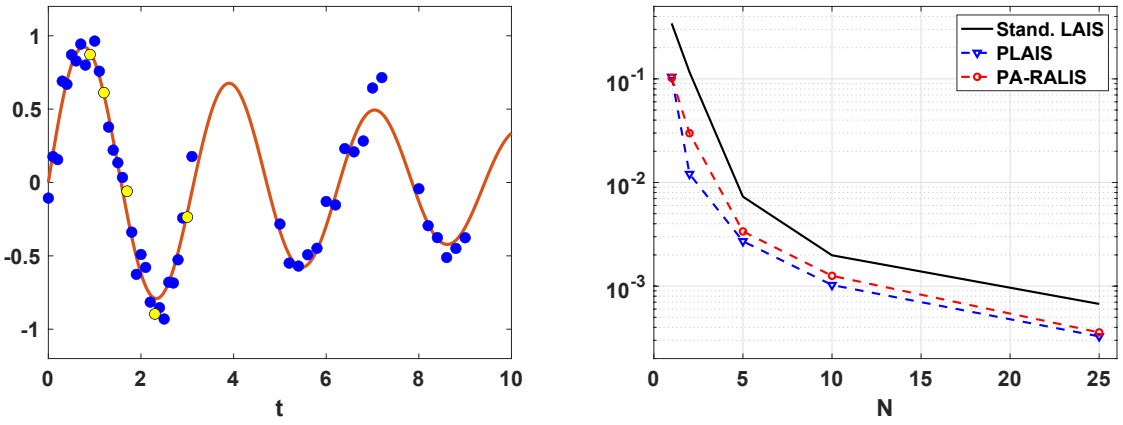
In this section, we consider a non-linear regression problem. This is a simplified version of astronomical models, e.g., see the second numerical example in [21]. We generate 50 observations, $\mathbf{y}_{\texttt{tot}} = \{y_i\}_{i=1}^{50}$, from the following observation model

$$y_i = \exp(-\alpha t_i)\sin(\beta t_i) + v_i$$

where the values α and β were fixed at 0.1 and 2, respectively. The error terms v_i were independently generated from a Gaussian, $\mathcal{N}(0, 0.1^2)$. For this model, we take $\mathbf{x} = [\alpha, \beta]^\top$ and set a uniform density over the rectangle $[0, 10] \times [0, 2\pi]$ as prior density for $\mathbf{x}$. Figure 3.2(a) shows the function $\exp(-\alpha t)\sin(\beta t)$ and some data generated according to the model. The goal is to investigate the use of partial posteriors in the LAIS framework when computing $\mathbb{E}[\mathbf{x}|\mathbf{y}_{\texttt{tot}}]$, $\text{var}[\mathbf{x}|\mathbf{y}_{\texttt{tot}}]$ (marginal variances) and $Z = p(\mathbf{y}_{\texttt{tot}})$. By using a very thin grid over the space, we are able to calculate the true values, obtaining $\mathbb{E}[\mathbf{x}|\mathbf{y}_{\texttt{tot}}] = [0.1, 2]^\top$, $\text{var}[\mathbf{x}|\mathbf{y}_{\texttt{tot}}] = [6.88 \cdot 10^{-5}, 8.38 \cdot 10^{-5}]^\top$ and $Z = 3.03 \cdot 10^{-15}$. We compute the MSE in estimating those quantities with the following methods: (a) LAIS, (b) PLAIS, and (c) PA-RLAIS.

For all the methods, the upper layer consists of N independent random walk Metropolis-Hastings (MH) algorithms with Gaussian proposals (the same for all the schemes). In the upper layer, PLAIS and PA-RLAIS differ from LAIS in that, instead of the full posterior, each of the N chains targets a different partial posterior (with the same number of data K_n for all n). In the lower layer, one sample was drawn from each of the Gaussian proposal pdf. The covariance matrix of all the Gaussian proposals was set to $\mathbf{C}_n = 2\mathbf{I}_2$ where $\mathbf{I}_2$ is a 2×2 unit matrix. In the lower layer, PA-RLAIS differs from LAIS and PLAIS, in that we do not need to draw samples, but all samples are recycled from the chains in the upper layer. In a first experiment, we test the values $N \in \{1, 2, 5, 10, 25\}$, and set $T = 20$, $K_n = 10$ for all n. The results (averaged over 10^3 runs) in terms of MSE are shown in Figure 3.2(b). We can already see the benefits of PLAIS and PA-RLAIS.
In a second experiment, we fix the number of total evaluations of the full-posterior to $E = 2000$. In this case, for any value of $N \in \{1, 2, 5, 10, 25, 50\}$ we change T, in order to keep constant the total number evaluations of the full-posterior (see Section 3.8). In each simulation the partial posteriors were created by choosing randomly K_n data, with $K_n \in \{5, 10\}$. Figure 3.2(a) depicts some data generated according to the model. The orange dots are the observations chosen to construct the partial posterior in one simulation with $K_n = 5$. Finally, in all the methods, the initial mean vectors were drawn from the prior, i.e., $\boldsymbol{\mu}_{n,0} \sim \mathcal{U}([0, 10] \times [0, 2\pi])$, for all n. The results are averaged over 500 independent simulations.

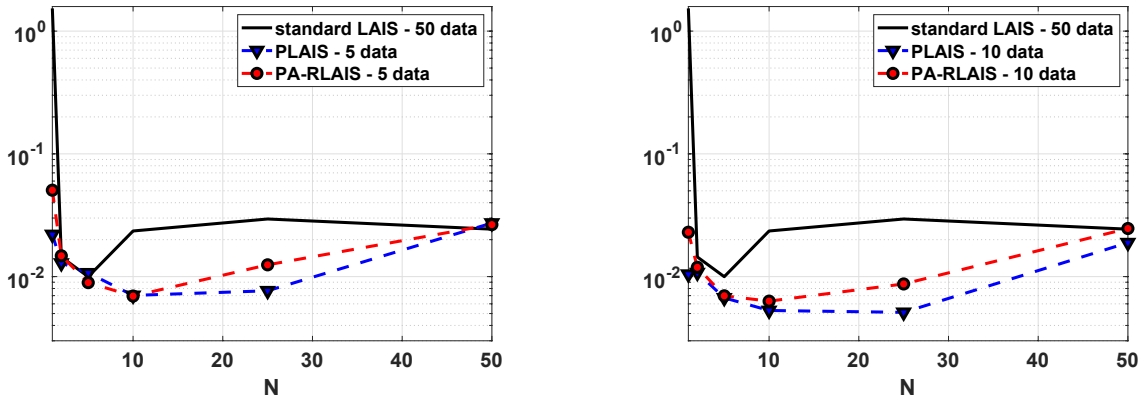
In Figure 3.3, we show the obtained results of this a second experiment. In both figures (a)-(b), we see the behavior of the MSE as N grows (and also T decreases, since we keep $E = 2000$ constant). The solid line corresponds to the standard LAIS implementation where we use all the data available for the computation of the likelihood in the upper layer. The dashed lines show the behavior of the errors when partial posteriors are considered in the upper layer. The left side shows the case $K_n = 5$ for all n, while, on the right side, we show $K_n = 10$ for all n. In both graphics, it can be seen that PLAIS and PA-RLAIS outperform the results of standard LAIS, for the values of N considered. Hence, in this simple example, using partial posteriors improves the performance of the algorithms. For all methods, the error tends to grow after certain optimal N (recall that T is also varying in this figure). However, the methods that use partial posteriors show better performance,



(a) (b)

Figure 3.2: Results corresponding to the experiment in Section 3.9.2. **(a)** The solid line is the function that defines the model, and the blue dots are the observations generated from it. The yellow dots represent an example of random subset of data used in a partial-posterior. **(b)** MSE versus N, with $T = 20$ and $K_n = 10$ for all $n = 1, ..., N$.

as compared to standard LAIS, when N increases, that is, when there is more number of shorter chains. This can be due to the fact that the partial posteriors are wider, and hence easier to explore in a small number of iterations. Also in both cases, the errors of PLAIS and PA-RLAIS are rather similar, although, as expected, PLAIS outperforms PA-RLAIS.



(a) (b)

Figure 3.3: Results corresponding to the experiment in Section 3.9.2. MSE obtained by the different algorithms for distinct numbers of data in the partial posteriors. Note that we keep fixed the total number of posterior evaluations to $E = 2000$. This means that as N grows, then T decreases (e.g., in standard LAIS we have $E = 2NT$); **(a)** with $K_n = 5$; **(b)** with $K_n = 10$.

3.9.3. HMC-LAIS vs HMC algorithms

For the next experiment, we consider $\bar{\pi}(\mathbf{x})$ which consists of an equally-weighted mixture of two Gaussian pdfs. The Gaussians pdfs are located at $[0,0]^\top$ and $[-4,4]^\top$, respectively. The covariance matrix of both is $\mathbf{\Sigma} = [4,3;3,4]$. Here, it is straightforward to calculate the true values for the quantities of interest: the expected value is $[-2,2]^\top$, the variances are $[8,8]$ and the covariance is -1. We test the performances of HMC-LAIS algorithms in estimating these quantities, i.e., expected value of $\bar{\pi}(\mathbf{x})$ (2 quantities), and covariance matrix of $\bar{\pi}(\mathbf{x})$ (3 quantities). The goal is to compare their performances against only using HMC algorithms. The error measure we employ is the averaged Mean Squared Error (MSE).

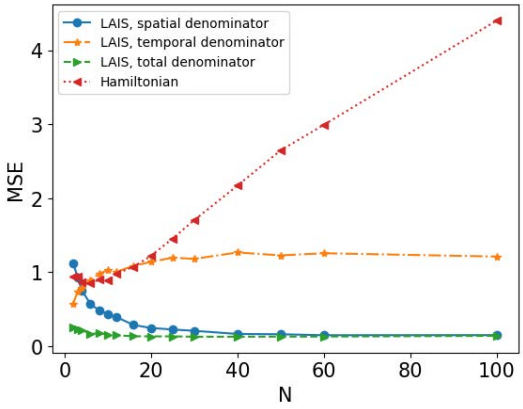
The computational budget is fixed to $E = 2400$ target evaluations. We consider HMC algorithms with kinetic energy using a Gaussian distribution with covariance matrix equal to $2\mathbf{I}$, and test the following values for step length and path length {(0.25,1),(0.5,1),(1,3),(1,5)}. In the lower layer, we also consider Gaussian proposals with covariance matrix equal to $\mathbf{C_n}$ =$2\mathbf{I}$. Here, we compare the performance of three deterministic-mixture weighting schemes: spatial, temporal and complete.
For setting the number of chains, N, and the number of iterations, T, we follow the same rules as for the previous experiment. We kept constant the product $NT = \frac{E}{2} = 1200$ and vary N within {2,3,4,6,8,10,12,16,20,25,30,40,50,60,100}. For a fair comparison, when we only consider HMC algorithms, the N chains were run for $2T$ iterations each (i.e. twice number of iterations than the HMC algorithms in the upper layer of the HMC-LAIS algorithms), so that the final number of target evaluations is $2NT = E = 2400$. The initial mean vectors were chosen uniformly within the square $[-10,10]^2$. The results were averaged over 500 independent simulations.
In Figure 3.4, we show the MSE of the HMC and HMC-LAIS algorithms, with three weighting schemes, as a function of N. Recall that, for every N, the HMC algorithms were run for twice number of iterations, i.e., they were run for $2T$ iterations, in order to have the same number of target evaluations. Each figure corresponds to a different choice of step and path lengths in the HMC algorithms.
First main observation. We can observe that the LAIS schemes (except some few specific cases) always outperform the HMC algorithms.
Second main observation. It is important to remark the excellent and robust performance provided by HMC-LAIS with the *complete* denominator, regardless the parameters of HMC chains (in the upper layer) used and the number of chains N. In fact, HMC-LAIS algorithm with complete denominator clearly outperforms the rest of techniques, providing the smallest error and remaining constant for all N and all HMC parameters.
Other considerations. The error of HMC is smallest when N is close to the minimum (i.e. when the chains are longer), and gets worse as N increases since, consequently, the chains become shorter and cannot explore properly the two modes. Interestingly, even in the best scenario, the results show that the error of HMC is always greater than the one

provided by HMC-LAIS algorithms with temporal and complete denominators. Namely, even when HMC works best, it is better to run it for half number of iterations and then use it within the LAIS framework with a temporal or complete denominator.

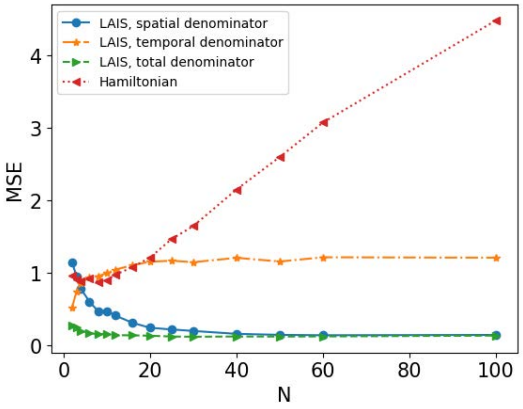
Spatial vs Temporal. The performance of the temporal and spatial denominators behave in an opposite manner. As expected, the error corresponding to the spatial denominator is worse when N is small. In fact, the greatest error is achieved always when N is minimum. As N increases, the performance greatly improves. It rapidly beats HMC and its performance matches that of the complete weighting scheme for large N. Conversely, in the temporal denominator, the best results are always achieved when N is minimum, since in this case, the chain length T is maximum. As N increases, the performance of the temporal denominator worsens, but in a slower fashion than the corresponding error of the HMC algorithms.

In this experiment, the spatial denominator seems to outperform the temporal denominator for more values of N. This means that the mixture of spatial proposals is usually better than the mixture of temporal proposals. For some value N_*, both weighting schemes provide the same results. Only for values $N \leq N_*$, the temporal denominator is better than the spatial denominator. Namely, if T is not sufficiently big ($T \leq \frac{E}{2N_*}$), the temporal denominator does not pay off, as compared to the spatial denominator. In fact, for $N > 50$, the spatial denominator can be considered as a compressed version of the complete denominator, i.e., it provides almost the same performance but with a smaller number of components (recall that the complete denominator has $\frac{E}{2} = 1200$ mixture components).

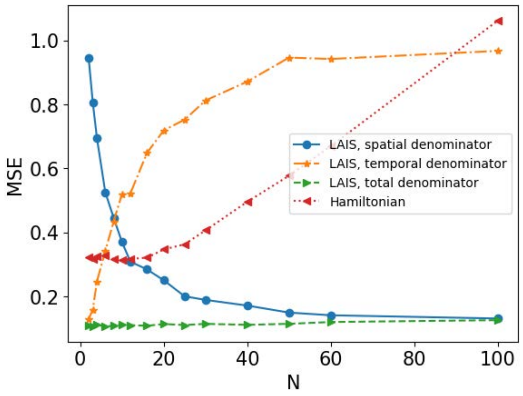
Compressed schemes. We have also tested the performance of compressed LAIS (CLAIS), where a compression technique is applied to the NT proposals from the upper layer (see Sect. 3.6). Here, we have run a clustering algorithm with $B \in \{3, 21, 50, 200\}$ clusters to obtain the compressed denominators. In Figure 3.5, we show the error of these schemes against the three previous weighting schemes and HMC. With the proposed compression scheme, we see that the performance is very close to that of the complete denominator and it is insensitive to the choice of number of clusters and N. For moderately low N, CLAIS outperforms LAIS with spatial denominator. However, as N increases, the spatial denominator matches the performance of CLAIS, i.e., the spatial denominator is also a very efficient way of compressing the NT proposals as discussed above. Finally, in Figure 3.6 we display the computation time of CLAIS versus the compression level η, which is $\eta = 0$ when there is no compression at all ($B = NT$, i.e. the maximum number of clusters), and $\eta = 1 - \frac{1}{NT}$ when we have $B = 1$ clusters.



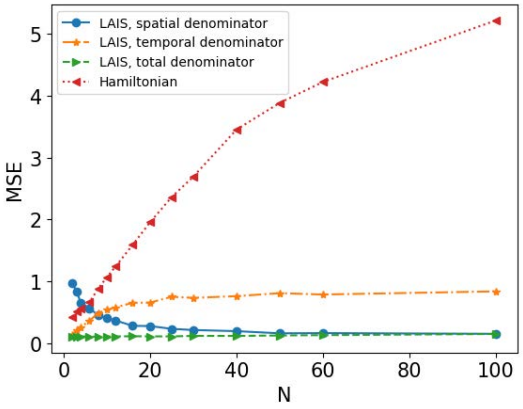
(a) HMC parameters (0.25,1)



(b) HMC parameters (0.5,1)

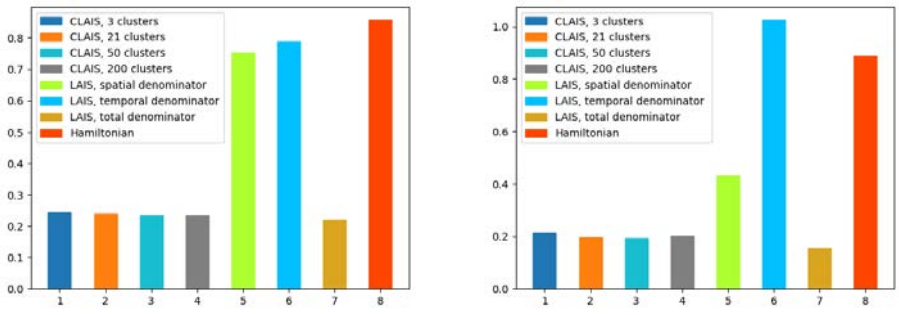


(c) HMC parameters (1,3)

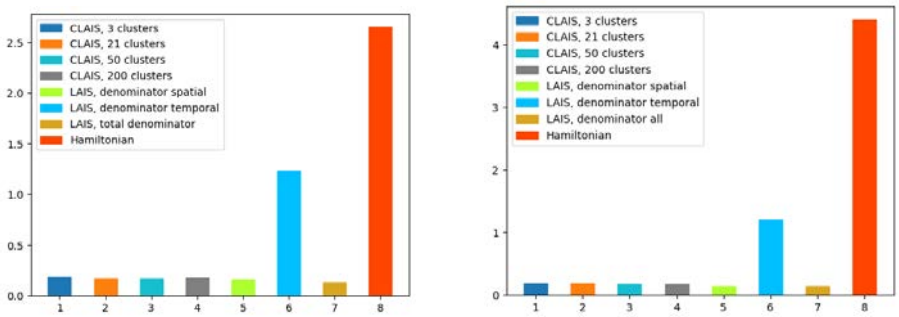


(d) HMC parameters (1,5)

Figure 3.4: Results corresponding to the experiment in Section 3.9.3. MSE in estimation obtained by HMC-LAIS and HMC versus N, with the same number of evaluations of the posterior $E = 2400$ (hence, the HMC chains have twice the length of the HMC chains used in the upper layer of HMC-LAIS). Each figure corresponds to a different choice of step and path lengths in the HMC algorithms.



(a) $N = 4$ (b) $N = 10$



(c) $N = 50$ (d) $N = 100$

Figure 3.5: Results corresponding to the experiment in Section 3.9.3. MSE of CLAIS with different values of $B \in \{3, 21, 50, 200\}$, compared with LAIS with different denominators and parallel HMC chains (with twice lengths with respect to the LAIS schemes, in order to have the same number of posterior evaluations, $E = 2400$, for all methods).

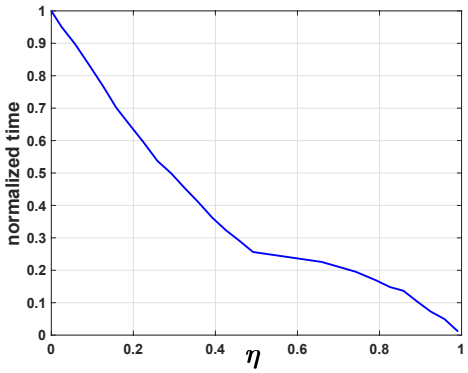


Figure 3.6: Results corresponding to the experiment in Section 3.9.3. Normalized computational time versus compression level η, where $\eta = 1 - \frac{B}{NT}$, and B is the number of clusters.

3.9.4. High-dimensional experiment

In order to be able to compare different schemes in a high-dimensional sampling problem, we need to know the groundtruth. For this reason, we assume a mixture of Gaussians as target pdf, i.e.,

$$\bar{\pi}(\mathbf{x}) = \frac{1}{3}\sum_{k=1}^{3} \mathcal{N}(\mathbf{x}; \boldsymbol{\nu}_k, \chi_k^2 \mathbf{I}_{D_X}), \quad \mathbf{x} \in \mathbb{R}^{D_X}, \tag{3.15}$$

where $\boldsymbol{\nu}_k = [\nu_{k,1}, \ldots, \nu_{k,D_X}]^\top$, for $k \in \{1, 2, 3\}$, with $\mathbf{I}_{D_X}$ being the $D_X \times D_X$ identity matrix and D_X is the dimension of the space. In this section, we vary the dimension of the state space in Eq. (3.15) considering $2 \leq D_X \leq 50$. Moreover, we set $\nu_{1,j} = -5$, $\nu_{2,j} = 6$, $\nu_{3,j} = 3$ for all $j = 1, ..., D_X$, and $\chi_k = 8$ for all $k \in \{1, 2, 3\}$. Note that the expected value of $\mathbf{X} \sim \bar{\pi}(\mathbf{x})$ is $E[X_j] = \frac{4}{3}$ for $j = 1, \ldots, D_X$. In order to study the performance of different Monte Carlo methods, we consider the problem of approximating this expected value. We apply HMC-LAIS considering $N = 100$ parallel chains of HMC in the upper layer, each chain with different parameters. The HMC chains require the selection of following parameters: a positive integer number of "leap-frog steps" Q, a positive number for the step size ζ and the covariance matrix of the Gaussian kinetic energy $\lambda^2 \mathbf{I}_{D_X}$ (where we set $\lambda = 10$). We select the two first parameters both randomly, for each chain and at each run: we select Q uniformly between 1 and 7 (it must be an integer), and $\zeta \in \mathcal{U}([0.01, 0.7])$. The proposal pdfs used in the lower layer, $q_{n,t}(\mathbf{x}|\boldsymbol{\mu}_{n,t}, \mathbf{C}_n)$, are Gaussian pdfs with covariance matrices $\mathbf{C}_n = \sigma^2 \mathbf{I}_{D_X}$ again with $\sigma = 10$. We also draw $M > 1$ (more than one samples) from each proposal in the upper layer. More precisely, we set $M = 19$ and the length of the chains $T = 100$ because, since $N = 100$, we have a total number of target evaluations of $E = (M + 1)NT = 2 \cdot 10^5$.

We compare HMC-LAIS with different benchmark schemes: **(a)** the standard PMC scheme [4], **(b)** N parallel independent MH chains (Par-MH), **(c)** and a Sequential Monte Carlo (SMC) scheme [24]. For a fair comparison, all the mentioned algorithms have been implemented in such a way that the number of total evaluations of the target is $E = 2 \cdot 10^5$ as in HMC-LAIS. Moreover, all the proposal pdfs involved in the experiments are Gaussians, with the same covariance matrices for all the techniques. The initial mean vectors in all techniques are selected randomly and independently as $\boldsymbol{\mu}_{n,0} \sim \mathcal{U}([-6, 6]^{D_X})$ for $n = 1, \ldots, N$.

The results are averaged over 10^3 independent runs. Figure 3.7 shows (in log-scale) the MSE in the estimation of $E[\mathbf{X}]$ as a function of the dimension D_X of the support space. We remark that we have kept fixed the number of total evaluations of the target $E = 2 \cdot 10^5$ for all the techniques. As expected, the performance of all the methods deteriorates as the dimension of the problem, D_X increases, since we maintain fixed the computational cost $E = 2 \cdot 10^5$. HMC-LAIS always provides the best results, i.e., obtaining the lower MSE values.

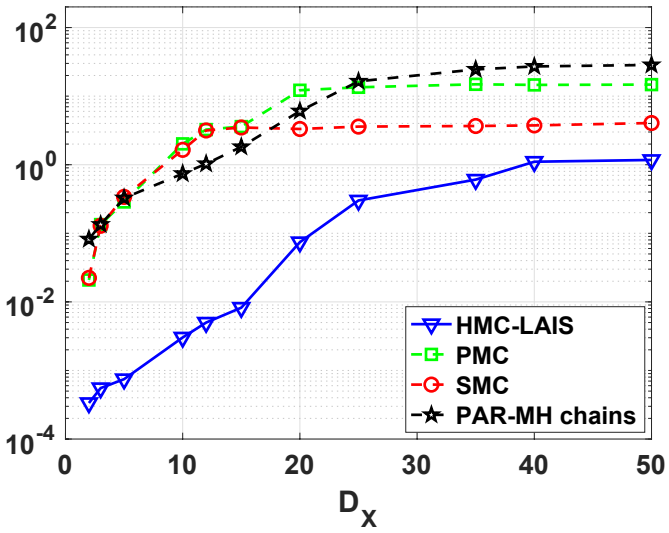


Figure 3.7: Results corresponding to the experiment in Section 3.9.4. MSE (in log-scale) versus the dimension of the space D_X, obtained by the different samplers, with the same total number of target evaluations $E = 2 \cdot 10^5$. Namely, we keep fixed the computational cost, that in HMC-LAIS means keeping fixed the parameters $N = 100$, $M = 19$ and $T = 100$ (for all D_X).

3.9.5. Parameter estimation in a chaotic system

In this section, we show that the use of Gibbs-LAIS can be useful in complex inference scenarios where sophisticated MCMC techniques seem to fail (see, for instance, [27] or [28]). We consider the problem of estimating parameters in a chaotic system, which is considered a very challenging framework in the literature (see, e.g., [14] or [27]). This is due to the very tight and sharp posteriors induced by this model. As an example, see the conditional posterior densities in Figure 3.8. The density in Figure 3.8(c) is extremely tight (resembling a delta function), so even sophisticated adaptive Monte Carlo techniques fail. This type of systems are often utilized for modeling the evolution of population sizes, for instance in ecology [28]. Specifically let us consider a logistic map [3] perturbed by multiplicative noise,

$$y_{k+1} = R\left[\, y_k \left(1 - \frac{y_k}{\Omega}\right)\right] \exp(\epsilon_k), \quad \epsilon_k \sim \mathcal{N}(0, \lambda^2), \quad k = 1, ..., K, \tag{3.16}$$

starting with $y_1 \sim \mathcal{U}([0, 1])$. The parameters $R > 0$ and $\Omega > 0$ are unknown and object of the inference. Hence, using the notation in this work, we have $\mathbf{x} = [R, \Omega]$. Let us assume that a sequence $\mathbf{y} = y_{1:K} = [y_1, \ldots, y_K]$ is observed and, for the sake of simplicity, let us consider that the standard deviation λ of the noise is known. The corresponding likelihood function is given by

$$L(\mathbf{y}|\mathbf{x}) = p(y_{1:K}|R, \Omega) = \prod_{k=1}^{K-1} p(y_{k+1}|y_k, R, \Omega),$$

where, denoting $b(y_k, R, \Omega) = R\left[\, y_k \left(1 - \frac{y_k}{\Omega}\right)\right]$, we have

$$p(y_{k+1}|y_k, R, \Omega) \propto \left|\frac{g(y_k, R, \Omega)}{y_{k+1}}\right| \exp\left(-\frac{\log\left(\frac{y_{k+1}}{g(y_k,R,\Omega)}\right)^2}{2\lambda^2}\right), \quad \text{if } b(y_k, R, \Omega) > 0,$$

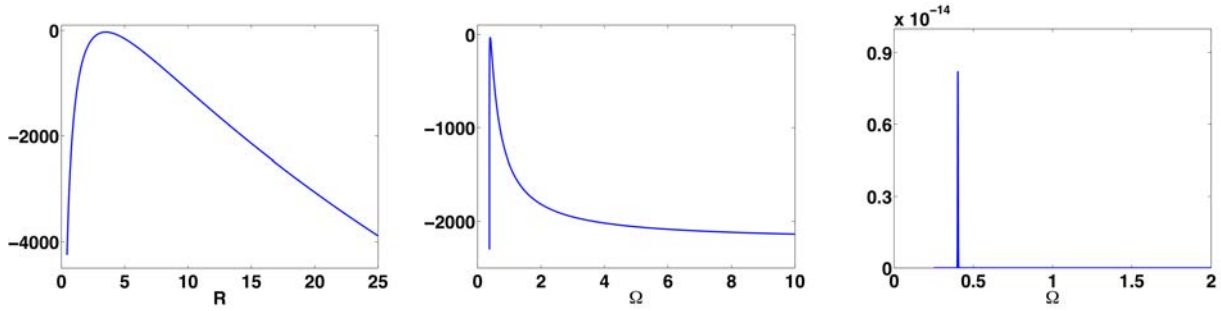
and $p(y_{k+1}|y_k, R, \Omega) = 0$, if $b(y_k, R, \Omega) \leq 0$. We set uniform priors, $R \sim \mathcal{U}([0, 10^4])$ and $\Omega \sim \mathcal{U}([0, 10^4])$, our goal is computing the mean of the bivariate posterior pdf, $\bar{\pi}(\mathbf{x}|\mathbf{y}) = p(R, \Omega|y_{1:K}) \propto p(y_{1:K}|R, \Omega)$, which represents the minimum mean square error estimator of the vector parameter $\mathbf{x} = [R, \Omega]$ (computing the MSE obtained by the different techniques).

We have generated artificial data $\mathbf{y} = y_{1:K}$, setting $R = 3.7$, $\Omega = 0.4$ and $K = 20$ (i.e., a trajectory of 20 values). We employ different values of standard deviation $\lambda = \{0.001, 0.005, 0.01, 0.05, 0.08, 0.1\}$ of the noise in the system (3.16) of the same order of magnitude considered in [27]. We apply a Gibbs-LAIS scheme where, for drawing from the full-conditional pdfs, we apply (within the Gibbs sampler) the so-called FUSS technique proposed in [22]. For simplicity, we consider a unique Gibbs chain ($N = 1$) in the upper layer with length $T = 25$ iterations, i.e., $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_T$. In the lower layer of Gibbs-LAIS scheme, we consider two-dimensional Gaussian proposals $q(\mathbf{x}|\boldsymbol{\mu}_t) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_t, \sigma_p^2 \mathbf{I}_2)$ with $\sigma_p = 1$ and $\mathbf{I}_2$ is the 2×2 identity matrix. We draw one sample from each proposal $q(\mathbf{x}|\boldsymbol{\mu}_t)$, hence we have $S = 25$ samples in the lower layer. Therefore, the total number of posterior evaluations of the Gibbs-LAIS scheme is $E = 25+25 = 50$. Since we have only one chain ($N = 1$), we use a temporal weighting scheme. We also apply the corresponding Gibbs-RLAIS with the same parameters (then $E = 25$), and also we perform a Gibbs-RLAIS but increasing the length of the Gibbs sampler to $T = 50$ (so that again $E = 50$). Finally, we compare the results with an MH-within-Gibbs approach with a Gaussian random walk proposal ($\sigma_p = 1$ again) for drawing from the full-conditionals, i.e., with $T = 50$ steps for the Gibbs samplers, in order to have $E = 50$ for a fair comparison. For the employed MCMC techniques, the initial states of the chains are chosen randomly from $\mathcal{U}([1, 5])$ for R and $\mathcal{U}([0.38, 1.5])$ for Ω.

The MSE in estimation obtained by the different techniques (averaged over 1000 independent runs) is given in Table 3.8. The Gibbs-LAIS schemes outperform clearly the MH-within-Gibbs approach. Moreover, Gibbs-RLAIS with $E = 25$ obtains very close results to Gibbs-LAIS, and Gibbs-RLAIS with $E = 50$ even outperforms Gibbs-LAIS when λ grows. Another remarkable advantage of employing the Gibbs-LAIS schemes is that one could easily approximating the marginal likelihood $Z = p(\mathbf{y}) = p(y_{1:K})$ in this problem, by computing the estimator $\widehat{Z}$ in (3.5). In this way, we could perform a model selection study. On the other hand, approximating Z by MH-within-Gibbs method is not a straightforward task [17].

Table 3.8: MSE in estimation of R and Ω, obtained by the different compared techniques.

			λ					
	E	Parameter	0.001	0.005	0.01	0.05	0.08	0.10
Gibbs-LAIS	50	R	0.0065	0.0067	0.0085	0.0125	0.0142	0.0681
		Ω	$4.97\ 10^{-5}$	$6.16\ 10^{-5}$	$4.18\ 10^{-5}$	$5.26\ 10^{-5}$	$6.33\ 10^{-5}$	$1.70\ 10^{-4}$
Gibbs-RLAIS	25	R	0.0082	0.0090	0.0089	0.0138	0.0160	0.0752
		Ω	$5.21\ 10^{-5}$	$6.22\ 10^{-5}$	$6.13\ 10^{-5}$	$4.22\ 10^{-5}$	$5.89\ 10^{-5}$	$1.82\ 10^{-4}$
Gibbs-RLAIS	50	R	0.0070	0.0069	0.0078	0.0126	0.0130	0.0547
		Ω	$5.01\ 10^{-5}$	$6.20\ 10^{-5}$	$5.75\ 10^{-5}$	$5.19\ 10^{-5}$	$6.08\ 10^{-5}$	$1.56\ 10^{-4}$
MH-within-Gibbs	50	R	0.6830	0.7264	0.7067	1.1631	1.3298	1.3293
		Ω	0.0373	0.0402	0.0423	0.0399	0.0471	0.0440



(a) Fixing $\Omega = 4$ (log-domain). (b) Fixing $R = 0.7$ (log-domain). (c) Natural domain.

Figure 3.8: Results corresponding to the experiment in Section 3.9.5. **(a)**-**(b)** Examples of conditional densities *in log-domain* with $\lambda = 0.1$, and considering $K = 20$ observations. **(a)** Fixing $\Omega = 4$. **(b)** Fixing $R = 0.7$. **(c)** The conditional pdf corresponding to the plot (b). Even advanced and adaptive MCMC techniques often fail in drawing samples from this kind of sharp/tight densities.

3.9.6. Experiment with COVID-19 data

This section is devoted to a model selection application. We consider the number of daily deaths caused by SAR-CoV-2 in Italy from 18 February 2020 to 6 July 2020 as the dataset. We denote the values of daily deaths as $\mathbf{y} = [y_1, \ldots, y_{D_Y}]^\top$. Let t_i denote the i-th day, we model each observation as

$$y_i = f(t_i) + e_i, \quad i = 1, \ldots, D_Y = 140,$$

where f is the function that we aim to approximate and e_i's are independent Gaussian realizations with zero means and variance σ_e^2. We consider the approximation of f at t as a weighted sum of M localized basis functions,

$$f(t) = \sum_{m=1}^{M} \rho_m \psi(t|\mu_m, h, \nu),$$

where $\psi(t|\mu_m, h)$ is m-th basis located at μ_m with bandwidth h. Let also be ν an index denoting the type of basis. We consider $M \in \{1, ..., D_Y\}$, then $1 \leq M \leq D_Y$. When $M = D_Y$,

the model becomes a Relevance Vector Machine (RVM), and the interpolation of all data points (maximum overfitting, with zero fitting error) is possible [2, 29]. We study 2 possible kinds of basis (i.e., $\nu = 1, 2$): Gaussian ($\nu = 1$), and Laplacian ($\nu = 2$). After fixing ν and M, we select the locations $\{\mu_m\}_{m=1}^M$ as a uniform grid in the interval $[1, D_Y]$ (recall that $D_Y = 140$). Hence, by knowing ν and M, the locations $\{\mu_m\}_{m=1}^M$ are given.

We define the vector of coefficients $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_M]^\top$. Let also $\boldsymbol{\Psi}$ be a $D_Y \times M$ matrix with elements $[\boldsymbol{\Psi}]_{i,m} = \psi(t_i|\mu_m, h)$ for $i = 1, \ldots, D_Y$ and $m = 1, \ldots, M$. Then, the observation equation in vector form is

$$\mathbf{y} = \boldsymbol{\Psi}\boldsymbol{\rho} + \mathbf{e},$$

where $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma_e^2 \mathbf{I}_{D_Y})$ is a $D_Y \times 1$ vector of noise, where $\mathbf{I}_{D_Y}$ is the $D_Y \times D_Y$ identity matrix. Therefore, the likelihood function will be

$$\ell(\mathbf{y}|\boldsymbol{\rho}, h, \sigma_e, \nu, M) = \mathcal{N}(\mathbf{y}|\boldsymbol{\Psi}\boldsymbol{\rho}, \sigma_e^2 \mathbf{I}_{D_Y}).$$

We assume a Gaussian prior density over the vector of coefficients $\boldsymbol{\rho}$, i.e., $g(\boldsymbol{\rho}|\lambda) = \mathcal{N}(\boldsymbol{\rho}|\mathbf{0}, \boldsymbol{\Sigma}_\rho)$, where $\boldsymbol{\Sigma}_\rho = \lambda \mathbf{I}_M$ and $\lambda > 0$. Therefore, the complete set of parameters to infer is $\{\boldsymbol{\rho}, \nu, M, h, \lambda, \sigma_e\}$. The conditional posterior of $\boldsymbol{\rho}$ given the rest of parameters is also Gaussian,

$$\bar{\pi}(\boldsymbol{\rho}|\mathbf{y}, \lambda, h, \sigma_e, \nu, M) = \frac{\ell(\mathbf{y}|\boldsymbol{\rho}, h, \sigma_e, \nu, M) g(\boldsymbol{\rho}|\lambda)}{p(\mathbf{y}|\lambda, h, \sigma_e, \nu, M)} = \mathcal{N}(\boldsymbol{\rho}|\boldsymbol{\mu}_{\rho|y}, \boldsymbol{\Sigma}_{\rho|y}),$$

and a likelihood marginalized w.r.t. $\boldsymbol{\rho}$ is available in closed-form,

$$p(\mathbf{y}|\lambda, h, \sigma_e, \nu, M) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \boldsymbol{\Psi}\boldsymbol{\Sigma}_\rho\boldsymbol{\Psi}^\top + \sigma_e^2 \mathbf{I}_{D_Y}). \tag{3.17}$$

For further details see [2, 29]. Now, we assume $g_\lambda(\lambda)$, $g_h(h)$, $g_\sigma(\sigma_e)$ are folded-Gaussian priors over h, λ, σ_e, defined on $\mathbb{R}_+ = (0, \infty)$ with location and scale parameters $\{0, 100\}$, $\{0, 400\}$ and $\{1.5, 9\}$, respectively. Then, we study the following posterior marginalized w.r.t. $\boldsymbol{\rho}$ and conditioned to μ, M,

$$\bar{\pi}(\lambda, h, \sigma_e|\mathbf{y}, \nu, M) = \frac{1}{p(\mathbf{y}|\nu, M)} p(\mathbf{y}|\lambda, h, \sigma_e, \nu, M) g_\lambda(\lambda) g_h(h) g_\sigma(\sigma_e),$$

Finally, we want to compute the marginal likelihood, i.e.,

$$p(\mathbf{y}|\nu, M) = \int_{\mathbb{R}_+^3} p(\mathbf{y}|\lambda, h, \sigma_e, \nu, M) g_\lambda(\lambda) g_h(h) g_\sigma(\sigma_e) d\lambda dh d\sigma_e. \tag{3.18}$$

Furthermore, assuming a uniform probability mass $p(M = i) = \frac{1}{D_Y}$ as prior over M, we have $p(M|\mathbf{y}, \nu) = \frac{p(\mathbf{y}|\nu, M) p(M)}{p(\mathbf{y}|\nu)} \propto \frac{1}{D_Y} p(\mathbf{y}|\nu, M)$. We can marginalize out M obtaining

$$p(\mathbf{y}|\nu) = \frac{1}{D_Y} \sum_{M=1}^{D_Y} p(\mathbf{y}|\nu, M), \quad \text{for} \quad \nu = 1, 2. \tag{3.19}$$

Considering also a uniform prior over ν, we can obtain the marginal posterior $p(\nu|\mathbf{y}) \propto \frac{1}{2}p(\mathbf{y}|\nu)$.

Goal. Our purpose is: (a) to make inference regarding the parameters of the model $\{\lambda, h, \sigma_e\}$, (b) approximate $Z = p(\mathbf{y}|\nu, M)$, (c) study the posterior $p(M|\mathbf{y}, \nu)$. We also study the marginal posterior $p(\nu|\mathbf{y})$ for $\nu = 1, 2$.

Methods. For approximating $p(\mathbf{y}|\nu, M)$, for $M = 1, \ldots, D_Y$, and $p(\nu|\mathbf{y})$, we first apply a Naive Monte Carlo (NMC) method with 10^4 samples [17]. We apply also a Gibbs-LAIS scheme with a MH-within-Gibbs sampler in the upper layer. More specifically, we employ an interpolative piecewise constant function as proposal in the MH scheme to draw from the full-conditionals (considering 2 internal steps) [22]. Hence, in the upper layer, we obtain a unique Markov chain ($N = 1$) of $\boldsymbol{\mu}_t = [\lambda_t, h_t, \sigma_{e,t}]$ for $t = 1, \ldots, T$. We set $T = 5000$, hence also 5000 samples drawn in the lower layer and used in estimators. The total number of evaluations of the posterior is $2T = 10^4$ for both, NMC and Gibbs-LAIS schemes.

Results. With both methods, We obtain that the MAP estimator of M is $M^* = 8$. In Figure 3.9, we show the fitting obtained with $M = 8$ bases and the parameter estimations provided by the Gibbs-LAIS scheme. Thus, a first conclusion is that the results obtained with models such as RVMs and Gaussian Processes (GPs) (both having $M = 140$ [2, 29]) can be approximated in a very good way with a much more scalable model, as our model here with only $M = 8$ [2, 29]. Regarding the marginal posterior $p(\nu|\mathbf{y})$, we can observe the results in Table 3.9. With the results provided by both schemes, we should prefer slightly the Laplacian basis. These considerations are reasonable after having a look at Figure 3.9.

Table 3.9: The approximate marginal posterior $p(\nu|\mathbf{y})$ with different techniques.

Method	$p(\nu = 1\|\mathbf{y})$	$p(\nu = 2\|\mathbf{y})$
NMC	0.4831	0.5169
Gibbs-LAIS	0.4930	0.5070

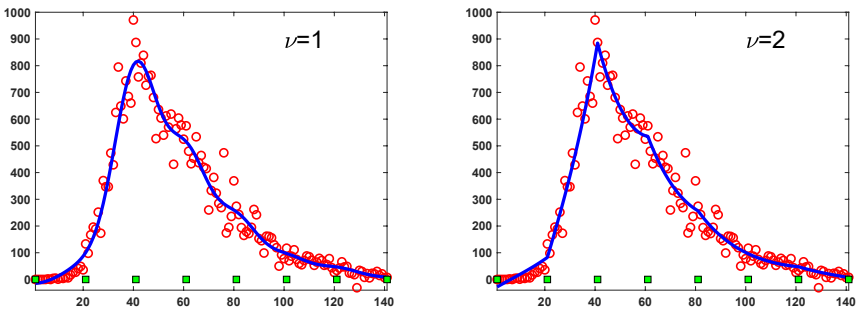


Figure 3.9: Results corresponding to the experiment in Section 3.9.6. Best fit with 8 bases with different types of basis, $\nu = 1, 2$. The circles represent the analyzed data and the squares show the positions of the bases.

3.10. Conclusions

We show the LAIS scheme is a flexible framework for designing efficient and robust AIS algorithms. Furthermore, we have introduced several enhancements in the LAIS framework in order to improve the performance and reduce the overall computational cost. Specifically, we have proposed that the MCMC algorithms in the upper layer address different partial posteriors (i.e., posteriors of subsets of data) to improve the mixing of the chains due to the data-tempering effect, and at the same time, reducing the costs of the upper layer. We have also studied the use of sophisticated MCMC algorithms, such as HMC and advanced Gibbs techniques, in the upper layer. These improvements allow the inference in very complex inference problems where other sophisticated techniques fail [27], as we have shown in Section 3.9.5. The proposed schemes are particularly useful to make inference with extremely concentrated posteriors, as shown in Figure 3.8(c), and where the computation of the marginal likelihood is also required. Moreover, the proposed methods provide also a clear improvement in high-dimensional inference spaces, as shown in Figure 3.7, obtaining at least a reduction of 25% in the estimation error.
Furthermore, we have designed a compression scheme for reduce the cost of the lower layer. Specifically, with the compression scheme, we can save more of the 70% of evaluations in the denominator of the IS weights (see Section 3.9.3). Numerous numerical experiments show that the proposed schemes outperform standard applications of the LAIS scheme and other benchmark algorithms. Interesting related theoretical considerations have been provided in the Appendices.
As future research lines, we consider that the automatic choice and the possible adaptation of the covariance matrices of the proposal densities in the lower layer are still open problems. Furthermore, the possible use of the MCMC samples also in the final estimators deserves additional studies.

3.11. Appendix

3.11.1. On the choice of the upper layer densities

Theoretical considerations: optimal invariant distribution in upper layer

Let us consider a hierarchical procedure which mimics the LAIS sample generation approach. For this purpose, we consider a single proposal pdf q in the lower layer defined by the mean $\boldsymbol{\mu} \in \mathbb{R}^{D_X}$ and scale matrix $\mathbf{C} \in \mathbb{R}^{D_X \times D_X}$, so that the proposal can be denoted as $q(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C})$, and it fulfills $q(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C}) = q(\mathbf{x} - \boldsymbol{\mu}|\mathbf{0}, \mathbf{C})$. This property is satisfied by relevant distributions such as Gaussian, Student's t and Laplace pdfs, for instance. The assumption is that the location parameter $\boldsymbol{\mu}$ is drawn exactly from the density $p(\boldsymbol{\mu})$. This is clearly a simplification since, with MCMC chains, we obtain correlated samples. Hence, the simplified LAIS generation procedure is given below:

1. Draw a possible location parameter $\boldsymbol{\mu}' \sim p(\boldsymbol{\mu})$.

2. Draw $\mathbf{x} \sim q(\mathbf{x}|\boldsymbol{\mu}', \mathbf{C})$.

Note that $p(\boldsymbol{\mu})$ plays the role of a prior pdf over the location parameter of the proposal density $q(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C})$. The sample $\mathbf{x}$ is distributed according to the following equivalent density,

$$\widetilde{q}(\mathbf{x}|\mathbf{C}) = \int_{\mathcal{X}} q(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C}) p(\boldsymbol{\mu}) d\boldsymbol{\mu} = \int_{\mathcal{X}} q(\mathbf{x} - \boldsymbol{\mu}|\mathbf{0}, \mathbf{C}) p(\boldsymbol{\mu}) d\boldsymbol{\mu}, \tag{3.20}$$

i.e., $\mathbf{x} \sim \widetilde{q}(\mathbf{x}|\mathbf{C})$. From Eq. (3.20) we can deduce the following considerations. The last expression in (3.20) is a convolution integral. Hence, considering the sum of two independent random variables

$$\mathbf{X} = \mathbf{Z} + \mathbf{M}, \tag{3.21}$$

where $\mathbf{Z} \sim q(\mathbf{x}|\mathbf{0}, \mathbf{C})$ (with $\boldsymbol{\mu} = 0$) and $\mathbf{M} \sim p(\boldsymbol{\mu})$, then $\mathbf{X}$ is distributed as $\widetilde{q}(\mathbf{x}|\mathbf{C})$ [30].
Now, let us consider the problem of finding the optimal density $p^*(\boldsymbol{\mu}|\mathbf{C})$ over the location parameter $\boldsymbol{\mu}$. In the LAIS scheme, the samples obtained by this procedure are then used in a self-normalized importance estimator. The variance of the IS weights is minimized when the proposal is exactly $\bar{\pi}(\mathbf{x}|\mathbf{y}_{\mathtt{tot}})$ [1, 30]. Therefore, the desirable scenario is to have $\widetilde{q}(\mathbf{x}|\mathbf{C}) = \bar{\pi}(\mathbf{x}|\mathbf{y}_{\mathtt{tot}})$. The optimal pdf depends on the chosen scale parameter $\mathbf{C}$ and since $q(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C}) = q(\mathbf{x} - \boldsymbol{\mu}|\mathbf{0}, \mathbf{C})$, as $\boldsymbol{\mu}$ is a location parameter, we can write

$$\bar{\pi}(\mathbf{x}|\mathbf{y}_{\mathtt{tot}}) = \int_{\mathcal{X}} q(\mathbf{x} - \boldsymbol{\mu}|\mathbf{0}, \mathbf{C}) p^*(\boldsymbol{\mu}|\mathbf{C}) d\boldsymbol{\mu}. \tag{3.22}$$

Equation (3.22) above can be rewritten in terms of the characteristic functions: $Q(\boldsymbol{\nu}|\mathbf{C}) = \int q(\mathbf{x}|\mathbf{0}, \mathbf{C}) e^{i\boldsymbol{\nu}^\top \mathbf{x}} d\mathbf{x}$, $P^*(\boldsymbol{\nu}|\mathbf{C}) = \int p^*(\mathbf{x}|\mathbf{C}) e^{i\boldsymbol{\nu}^\top \mathbf{x}} d\mathbf{x}$, and $\bar{\Pi}(\boldsymbol{\nu}) = \int \bar{\pi}(\mathbf{x}|\mathbf{y}_{\mathtt{tot}}) e^{i\boldsymbol{\nu}^\top \mathbf{x}} d\mathbf{x}$, where $\boldsymbol{\nu} \in \mathbb{R}^{D_X}$. The characteristic function of $\mathbf{X}$ is the product of characteristic functions of $\mathbf{Z}$ and $\mathbf{M}$. Hence, in some cases, the optimal invariant pdf in the upper layer has the following characteristic function,

$$P^*(\boldsymbol{\nu}|\mathbf{C}) = \frac{\bar{\Pi}(\boldsymbol{\nu})}{Q(\boldsymbol{\nu}|\mathbf{C})}. \tag{3.23}$$

In a general case, it is not possible to determine analytically the expression of the optimal pdf $p^*(\boldsymbol{\mu}|\mathbf{C})$, and thus, other practical choices must be considered, as discussed below.

Practical choices of the invariant distribution in the upper layer

Here, we discuss some practical selection of $p(\boldsymbol{\mu})$. First of all, from Eq. (3.21), we can obtain the following relevant considerations for this purpose:

1. $\mathrm{E}[\mathbf{X}] = \mathrm{E}[\mathbf{Z}] + \mathrm{E}[\mathbf{M}] = 0 + \mathrm{E}[\mathbf{M}]$, i.e., the expected value of the equivalent proposal $\widetilde{q}$ is equal to the expected value of the density $p(\boldsymbol{\mu})$ in the upper layer.

2. $\text{Var}[\mathbf{X}] = \text{Var}[\mathbf{Z}] + \text{Var}[\mathbf{M}] \geq \text{Var}[\mathbf{M}]$, where $\text{Var}[\cdot]$ returns the elements in the diagonal of the covariance matrix and, the inequality $\geq$ is applied to each element in the diagonal. Namely, the variances of each component of the equivalent proposal $\widetilde{q}$ are greater or equal to the variances of each component of the density $p(\boldsymbol{\mu})$ in the upper layer.

Thus, the equivalent density $\widetilde{q}(\mathbf{x}|\mathbf{C})$ has the same expected value and a bigger variance with respect to the density $p(\boldsymbol{\mu})$.

Consideration about the optimal pdf $p^*(\boldsymbol{\mu})$. Given Eq. (3.22) and the observations above, we can deduce that the optimal pdf $p^*(\boldsymbol{\mu})$ will have the same mean as the posterior, and it will have lighter tails than the posterior $\bar{\pi}$ (i.e., p^* is more "concentrated" than $\bar{\pi}$).

A possible choice of $p(\boldsymbol{\mu})$ in the upper layer. In practice, we cannot employ the optimal density $p^*(\boldsymbol{\mu})$. However, the choice $p(\boldsymbol{\mu}) = \bar{\pi}(\boldsymbol{\mu}|\mathbf{y}_{\texttt{tot}})$ provides an equivalent proposal with the same mean as the posterior, but with heavier tails. This is a relevant property: indeed, it avoids infinite variance estimators (see example 1 in [17]) and, as a consequence, this is the reason why this choice provides good performance in practice [20]. It can be shown that, in this case, the equivalent proposal is the kernel density estimator of the posterior (for a fixed optimal choice of $\mathbf{C}$). However, with a large amounts of data, evaluating the posterior can be very costly, so that the upper layer can require too much computational time. Furthermore, it is common that $\pi(\mathbf{x}|\mathbf{y}_{\texttt{tot}})$ is highly concentrated in some regions, so the MCMC algorithms in the upper layer can suffer from bad mixing. For these reasons, we provide the enhancements described in this work.

Standard tempering and anti-tempering

One idea for solving the second issue above, i.e., the bad mixing of the MCMC chains when $\pi(\mathbf{x}|\mathbf{y}_{\texttt{tot}})$ is highly concentrated, is the so-called *tempering*. Roughly speaking, tempering is a technique used to artificially change the scale of the target density. It is commonly used in order to improve the exploration of the posterior support in optimization, MCMC and IS [7, 26]. For instance, taking $p(\boldsymbol{\mu}) \propto \pi(\boldsymbol{\mu}|\mathbf{y}_{\texttt{tot}})^{\beta}$ with $0 < \beta < 1$ as the target density can be useful if $\bar{\pi}$ concentrates in a small region that is not easy to discover. The β is usually referred to as the (inverse) temperature parameter. More generally, a temperature schedule is a sequence of tempered posteriors ending with $\bar{\pi}$. A common choice is the geometric path between prior and posterior $\bar{\pi}_{\beta_n}(\mathbf{x}|\mathbf{y}_{\texttt{tot}}) \propto \pi(\mathbf{x}|\mathbf{y}_{\texttt{tot}})^{\beta_n} g(\mathbf{x})^{1-\beta_n} = L(\mathbf{y}_{\texttt{tot}}|\mathbf{x})^{\beta_n} g(\mathbf{x})$, for a sequence $0 = \beta_0 < \beta_1 < \cdots < \beta_N = 1$, such $\bar{\pi}_{\beta_0}(\mathbf{x}|\mathbf{y}_{\texttt{tot}}) = g(\mathbf{x})$ (i.e., the prior pdf over $\mathbf{x}$) and $\bar{\pi}_{\beta_N}(\mathbf{x}|\mathbf{y}_{\texttt{tot}}) = \bar{\pi}(\mathbf{x}|\mathbf{y}_{\texttt{tot}})$. Note that the tempered posterior has a powered, less informative (i.e., wider) likelihood.
Therefore, in order to improve the exploration of the posterior support, one possibility consists in taking $p_n(\boldsymbol{\mu}) = \bar{\pi}_{\beta_n}(\mathbf{x}|\mathbf{y}_{\texttt{tot}})$ in the upper layer.
Anti-tempering. An important point is that, in LAIS, one could set $\beta \leq 1$ in order to

foster the mixing of the chains, but also we can choose some $\beta > 1$ since, theoretically, the optimal pdf $p^*(\boldsymbol{\mu})$ is more "concentrated" than the posterior $\bar{\pi}$ (as described above). In any case, with a standard tempering strategy (using an auxiliary parameter β), we only solve one of the two issues pointed out in the rest of the work: improving the exploration of the posterior support. The cost of evaluating a tempered posterior $\bar{\pi}_{\beta_n}(\mathbf{x})$ is the same as the cost of evaluating the non-tempered posterior $\bar{\pi}$. An alternative to the standard tempering procedure is the so-called *data tempering*, which reduces also the evaluation cost by the use of the partial posteriors.

Hierarchical interpretation of the random walk Metropolis-Hastings (MH) algorithm

Consider a target density $\pi(\mathbf{x}) \propto \bar{\pi}(\mathbf{x})$ and a random-walk proposal pdf $q(\mathbf{x}|\mathbf{x}_{t-1}, \mathbf{C}) = q(\mathbf{x} - \mathbf{x}_{t-1}|\mathbf{0}, \mathbf{C})$, where $\mathbf{x}_{t-1}$ the current state of the chain and $\mathbf{C}$ is a covariance matrix. One transition of the MH algorithm is summarized by 1. Draw $\mathbf{x}'$ from a proposal pdf $q(\mathbf{x}|\mathbf{x}_{t-1}, \mathbf{C})$. 2. Set $\mathbf{x}_t = \mathbf{x}'$ with probability

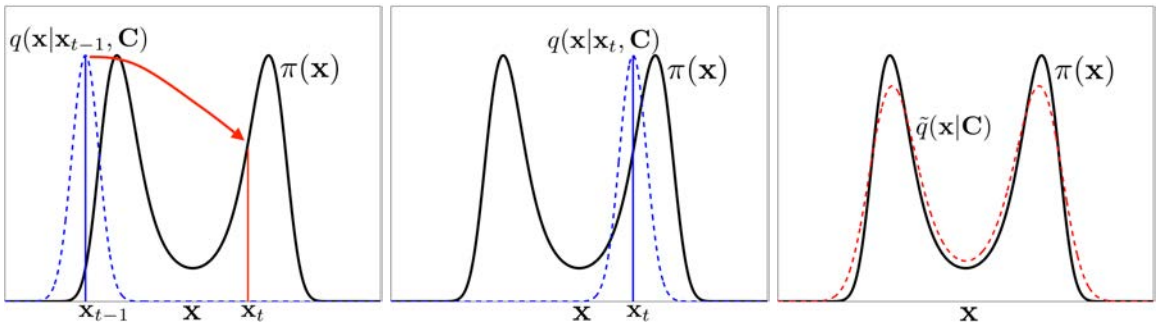
$$\alpha = \min\left[1, \frac{\pi(\mathbf{x}')\, q(\mathbf{x}_{t-1}|\mathbf{x}', \mathbf{C})}{\pi(\mathbf{x}_{t-1})\, q(\mathbf{x}'|\mathbf{x}_{t-1}, \mathbf{C})}\right]$$

otherwise set $\mathbf{x}_t = \mathbf{x}_{t-1}$ (with probability $1-\alpha$). There are two well-known general classes of proposal pdf: independent proposal q (independent from the current state), and random walk proposal, $q(\mathbf{x}|\mathbf{x}_{t-1}, \mathbf{C})$, as previously considered. The use of a random walk proposal $q(\mathbf{x} - \mathbf{x}_{t-1}|\mathbf{0}, \mathbf{C})$ is often preferred due to its explorative behavior, since it relocates the proposal at the current state of the chain at each iteration. See Figure 3.10(a)-(b), for an example. As a consequence, this approach is more robust with respect to the choice of the tuning parameters. Below, we provide some further arguments explaining the success of the random walk approach.

We provide a hierarchical interpretation in the same fashion on LAIS. Let us assume a "burn-in" length $T_b - 1$. Hence, considering an iteration $t \geq T_b$, we can assert $\mathbf{x}_t \sim \bar{\pi}(\mathbf{x})$. It implies that the random walk generating process is equivalent, for $t \geq T_b$, to the following hierarchical procedure: (a) draw a location parameter $\boldsymbol{\mu}'$ from $\bar{\pi}(\boldsymbol{\mu})$, (b) draw $\mathbf{x}'$ from $q(\mathbf{x}|\boldsymbol{\mu}', \mathbf{C})$. Therefore, for $t \geq T_b$, the probability of proposing a new sample (i.e., the equivalent proposal) can be written as

$$\begin{aligned}\widetilde{q}_{MH}(\mathbf{x}|\mathbf{C}) &= \int_{\mathcal{X}} q(\mathbf{x}|\mathbf{x}_{t-1}, \mathbf{C})\, \bar{\pi}(\mathbf{x}_{t-1})\, d\mathbf{x}_{t-1}, \\ &= \int_{\mathcal{X}} q(\mathbf{x} - \mathbf{x}_{t-1}|\mathbf{0}, \mathbf{C})\, \bar{\pi}(\mathbf{x}_{t-1})\, d\mathbf{x}_{t-1}, \qquad \text{for } t \geq T_b, \end{aligned} \tag{3.24}$$

since $\mathbf{x}_{t-1} \sim \bar{\pi}(\mathbf{x}_{t-1})$ after a burn-in period, $t \geq T_b$, and $\mathbf{x}_{t-1}$ represents the location parameter of q. The function $\widetilde{q}_{MH}(\mathbf{x}|\mathbf{C})$ is an equivalent independent proposal pdf corresponding to a random walk generating process within an MCMC method (after the "burn-in" period). See Figure 3.10(c) for an example of $\widetilde{q}_{MH}$.



(a) (b) (c)

Figure 3.10: Graphical representation of the equivalent proposal of a random-walk proposal in a MH method. A bimodal target pdf $\pi(\mathbf{x})$ is shown in solid line. The proposal densities are depicted in dashed lines. **(a)** A proposal pdf $q(\mathbf{x}|\mathbf{x}_{t-1}, \mathbf{C}) = q(\mathbf{x} - \mathbf{x}_{t-1}|\mathbf{0}, \mathbf{C})$ at the iteration $t-1$, and the next state of the chain $\mathbf{x}_t$. **(b)** The proposal pdf $q(\mathbf{x}|\mathbf{x}_t, \mathbf{C}) = q(\mathbf{x} - \mathbf{x}_t|\mathbf{0}, \mathbf{C})$ at the t-th iteration. **(c)** The equivalent independent proposal pdf $\widetilde{q}_{MH}(\mathbf{x}|\mathbf{C})$ is represented in dashed line.

Clearly, this interpretation has no direct implications for practical purposes, since we are not able to draw directly form the target $\bar{\pi}$. However, it is useful for clarifying the main advantage of the random walk approach, i.e., that the equivalent proposal $\widetilde{q}_{MH}$ is a better choice than an independent proposal roughly tuned by the user with non-optimal parameters. In fact, as an example, Eq. (3.24) ensures that the equivalent proposal $\widetilde{q}_{MH}(\mathbf{x}|\mathbf{C})$ has a fatter tails than the target $\bar{\pi}$. Indeed, the random walk generating procedure includes indirectly certain information about the target: denoting $\mathbf{X} \sim \tilde{q}_{MH}(\mathbf{x}|\mathbf{C})$, $\mathbf{Z} \sim q(\mathbf{x}|\mathbf{0}, \mathbf{C})$ and $\mathbf{M} \sim \bar{\pi}(\mathbf{x})$, we have

$$E[\mathbf{X}] = E[\mathbf{M}], \quad \mathbf{\Sigma}_X = \mathbf{C} + \mathbf{\Sigma}_M,$$

where $E[\mathbf{M}]$ and $\mathbf{\Sigma}_M$ are the mean and covariance matrix of the target pdf $\bar{\pi}(\mathbf{x})$.

Bibliography

[1] O. D. Akyildiz and J. Miguez. Convergence rates for optimised adaptive importance samplers. *Statistics and Computing*, 31(12):1–17, 2021.

[2] C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York (NY) USA, 2006.

[3] A. Boyarsky and P. Góora. *Law of Chaos*. Birkhäuser, Boston (USA), 1997.

[4] O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.

[5] J. Corander, M. Ekdahl, and T. Koski. Parallel interacting MCMC for learning of topologies of graphical models. *Data Mining and Knowledge Discovery*, 17(3):431–456, 2008.

[6] J. M. Cornuet, J. M. Marin, A. Mira, and C. P. Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, December 2012.

[7] D. J. Earl and M. W. Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.

[8] Y. El-Laham, L. Martino, V. Elvira, and M. F. Bugallo. Efficient adaptive multiple importance sampling. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019.

[9] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Improving population Monte Carlo: Alternative weighting and resampling schemes. *Signal Processing*, 131:77–91, 2017.

[10] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Generalized Multiple Importance Sampling. *Statistical Science*, 34(1):129–155, 2019.

[11] W. R. Gilks, N. G. Best, and K. K. C. Tan. Adaptive Rejection Metropolis Sampling within Gibbs Sampling. *Applied Statistics*, 44(4):455–472, 1995.

[12] W. R. Gilks, G. O. Roberts, and E. I. George. Adaptive direction sampling. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 43(1):179–189, 1994.

[13] W. R. Gilks and P. Wild. Adaptive Rejection Sampling for Gibbs Sampling. *Applied Statistics*, 41(2):337–348, 1992.

[14] F. Hartig and C. F. Dormann. Does model-free forecasting really outperform the true model? *Proceedings of the National Academy of Sciences (PNAS)*, 110(42):E3975, 2013.

[15] F. Liang, C. Liu, and R. Caroll. *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*. Wiley Series in Computational Statistics, Padstow (Cornwall), England, 2010.

[16] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, New York (NY), USA, 2004.

[17] F. Llorente, L. Martino, D. Delgado, and J. Lopez-Santiago. Marginal likelihood computation for model selection and hypothesis testing: an extensive review. *(to appear) SIAM Review*, 2022.

[18] L. Martino and V. Elvira. Compressed Monte Carlo with application in particle filtering. *Information Sciences*, 553:331–352, 2021.

[19] L. Martino, V. Elvira, D. Luengo, and J. Corander. An adaptive population importance sampler: Learning from the uncertanity. *IEEE Transactions on Signal Processing*, 63(16):4422–4437, 2015.

[20] L. Martino, V. Elvira, D. Luengo, and J. Corander. Layered adaptive importance sampling. *Statistics and Computing*, 27:599–623, 2017.

[21] L. Martino, F. Llorente, E. Curbelo, J. Lopez-Santiago, and J. Miguez. Automatic tempered posterior distributions for Bayesian inversion problems. *Mathematics*, 9(7), 2021.

[22] L. Martino, H. Yang, D. Luengo, J. Kanniainen, and J. Corander. A fast universal self-tuned sampler within Gibbs sampling. *Digital Signal Processing*, 47:68–83, 2015.

[23] R. Meyer, B. Cai, and F. Perron. Adaptive rejection Metropolis sampling using Lagrange interpolation polynomials of degree 2. *Computational Statistics and Data Analysis*, 52(7):3408–3423, March 2008.

[24] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[25] R. Neal. *MCMC Using Hamiltonian Dynamics*. Chapter 5 of the Handbook of Markov Chain Monte Carlo, Edited by Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng; Chapman and Hall/CRC Press, London, England, 2011.

[26] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

[27] C. T. Perretti, S. B. Munch, and G. Sugihara. Model-free forecasting outperforms the correct mechanistic model for simulated and experimental data. *Proceedings of the National Academy of Sciences (PNAS)*, 110(13):5253–5257, 2013.

[28] C. T. Perretti, S. B. Munch, and G. Sugihara. Reply to Hartig and Dormann: The true model myth. *Proceedings of the National Academy of Sciences (PNAS)*, 110(42):E3976–E3977, 2013.

[29] C. E. Rasmussen. *Gaussian Processes for Machine Learning*. the MIT Press, Cambridge (Massachusetts), USA, 2006.

[30] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, New York (NY), USA, 2004.

[31] D. Rudolf and B. Sprungk. On a Metropolis-Hastings importance sampling estimator. *Electronic Journal of Statistics*, 14(1):857–889, 2020.

[32] I. Schuster and I. Klebanov. Markov Chain Importance Sampling? A highly efficient estimator for MCMC. *Journal of Computational and Graphical Statistics*, pages 1–9, 2020.

[33] Steven L. Scott, Alexander W. Blocker, Fernando V. Bonassi, Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bayes and big data: The consensus Monte Carlo algorithm. In *EFaBBayes 250th conference*, volume 16, 2013.

[34] E. Veach and L. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. *In SIGGRAPH 1995 Proceedings*, pages 419–428, 1995.

4. ADAPTIVE QUADRATURE SCHEMES FOR BAYESIAN INFERENCE VIA ACTIVE LEARNING

In *IEEE Access*, Volume 8, 208462-208483 (2020)

F. Llorente*, L. Martino⋆, V. Elvira†, D. Delgado*, J. López-Santiago*

* Universidad Carlos III de Madrid, Léganes, Madrid, Spain.

† The University of Edinburgh, Edinburgh, United Kingdom.

⋆ Universidad Rey Juan Carlos I, Móstoles, Madrid, Spain.

Abstract

We propose novel adaptive quadrature schemes based on an active learning procedure. We consider an interpolative approach for building a surrogate posterior density, combining it with Monte Carlo sampling methods and other quadrature rules. The nodes of the quadrature are sequentially chosen by maximizing a suitable acquisition function, which takes into account the current approximation of the posterior and the positions of the nodes. This maximization does not require additional evaluations of the true posterior. We introduce two specific schemes based on Gaussian and Nearest Neighbors bases. For the Gaussian case, we also provide a novel procedure for fitting the bandwidth parameter, in order to build a suitable emulator of a density function. With both techniques, we always obtain a positive estimation of the marginal likelihood (a.k.a., Bayesian evidence). An equivalent importance sampling interpretation is also described, which allows the design of extended schemes. Several theoretical results are provided and discussed. Numerical results show the advantage of the proposed approach, including a challenging inference problem in an astronomic dynamical model, with the goal of revealing the number of planets orbiting a star.

Keywords: Numerical integration; emulation; Monte Carlo methods; Bayesian quadrature; experimental design; active learning.

4.1. Introduction and brief overview

In this work, we consider the approximation of intractable integrals of type

$$I = \int_{\mathcal{X}} f(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x},$$

where $f(\mathbf{x})$ is a generic integrable function and $\bar{\pi}(\mathbf{x})$ is a probability density function (pdf). These integrals usually appear in Bayesian inference problems where $\bar{\pi}(\mathbf{x})$ represents the posterior distribution of the variable of interest given the observed data. In the next subsections, we briefly review several approaches presented in the literature, which are related to the methodology presented this work.

4.1.1. Main families of quadrature methods

With the term *numerical integration*, we refer to a broad family of algorithms for calculating definite integrals, and by extension, the term is also used to describe the numerical solution of differential equations. Although exact analytical solutions to integrals are always desirable, such "unicorns" are rarely available, specially in real-world systems. Indeed, many applications in signal processing, statistics, and machine learning inevitably require the approximation of intractable integrals [9, 65, 52]. In particular, Bayesian methods need the computation of posterior expectations which, generally, are analytically intractable [65, 46]. The term numerical quadrature (or simply quadrature) is employed as a synonym for numerical integration [9]. More specifically, a quadrature formula is often stated as a weighted sum of integrand evaluations at specified points (a.k.a., nodes or knots) within the domain of integration.

Deterministic quadratures. A first family of numerical integration methods are the deterministic quadrature rules. A subclass within this family is the Newton-Cotes quadrature rules [9]. The Newton-Cotes formulas are based on evaluating the integrand at equally spaced nodes and are obtained by substituting the integrand function with a corresponding polynomial interpolation. Smaller approximation errors can often be obtained by using the Gaussian quadratures, where the nodes are optimally placed [9, 38, 29]. However, their applicability is restricted to certain particular cases.

Monte Carlo (MC) methods. A second family is formed by stochastic quadrature rules based on MC sampling methods [65, 46], such as Markov chain Monte Carlo (MCMC) and importance sampling algorithms. In this framework, the nodes of the quadrature rules are randomly chosen. However, the resulting estimators often have a high variance, specially when the dimension of the problem grows.

Variance Reduction. A third family, formed by the variance reduction techniques [56, 65], combines elements of the first two classes. In order to reduce the variance of the corresponding Monte Carlo estimators, deterministic procedures are included within the sampling algorithms, e.g., conditioning, stratification, antithetic sampling, and control variates [56]. Other interesting examples are the Riemann-based approximations which are combinations of a Riemann quadrature and random sampling [65, Chapter 4.3]. The Quasi-Monte Carlo (QMC) algorithms can be also included in this family. In QMC, deterministic sequences of points are generated (based on the concept of low-discrepancy) and then used as nodes of the corresponding quadrature [52]. Several other combinations of the previous classes above, mixing determinism with random sampling schemes, can be found in the literature [18, 2, 39].

Bayesian quadrature (BQ). The BQ framework represents a fourth approach which employs Gaussian Process (GP) regression algorithms for approximating the integrand function (and, as a consequence, the resulting integral as well) [54, 36, 63]. In the last years, this approach has raised the interest of several authors. One problem with this approach is that, in some cases, a negative estimation of the marginal likelihood can be obtained. Some possible solutions have been proposed, although they are quite complex based on

successive approximations [55, 26]. In this work, we provide two novel and much simpler alternatives for solving this issue. Moreover, unlike this work, most contributions in BQ literature focus on the GP approximation of the function $f(\mathbf{x})$ [55, 26, 7], although other papers on BQ describe quite general frameworks where $f(\mathbf{x})$ can contain the likelihood or $\pi(\mathbf{x})$ [54, 36, 63]. A connection between classical quadratures and BQ can be found in [34]. Finally, theoretical guarantees for adaptive BQ schemes can be found in the insightful work of [31].

4.1.2. Emulation of complex models

Many Bayesian inference problems involve the evaluation of computationally intensive models, because of **(i)** the use of particularly complex systems or **(ii)** a large number of available data (or both). To overcome this issue, one possible strategy consists in replacing the true model by a surrogate model (a.k.a. an *emulator*), that could be also adaptively improved [10, 57, 72]. Then, Bayesian inference is carried out on this approximate, cheaper model.

Use of the emulator. The emulator can be applied mainly in three different ways. **(a)** One possibility is to apply MC sampling methods considering the surrogate model as the target pdf [77, 62]. This is used to speed up the MC algorithms. **(b)** In order to improve the efficiency of MC estimators, a second option is to use the emulator as a proposal density within an MC technique, as we discuss in Section 4.1.3 [22, 21, 43]. **(c)** A third possibility is to replace the true posterior with the emulator in the integrals of interest, and computing them [54, 36, 63]. Here, we mainly focus on the last approach, also combining it with MC methods (and other quadrature rules).

Construction of the emulator. In the literature, the surrogate model is often built by using a regression algorithm, like a GP model or similar techniques [11, 75]. This probabilistic approach provides also uncertainty quantification that is used for estimating the approximation error and adapting the emulator [71]. Sometimes, the approximation regards only some part of the model or is applied in a different domain (as the log-domain) [6, 15, 32, 30]. Other authors employ density estimation techniques for building the surrogate model, and then using it as a proposal density within MC algorithms [13, 27, 45] or for replacing the true posterior (again within MC methods) [16].

4.1.3. Interpolative proposal densities within Monte Carlo schemes

The first use of an interpolative procedure for building a proposal density is ascribable to the adaptive rejection sampling schemes [22, 28, 24, 47]. The proposal is formed by polynomial pieces (constant, linear, etc.). Several works have proposed the use of interpolative proposal densities within MCMC algorithms [21, 50, 48, 49]. For more details, see also [46, Chapters 4 and 7]. Their use within an importance sampling scheme is considered in [20]. The adaptation is carried out considering different statistical tests, by measuring the discrepancy between the emulator and the posterior [43].

The conditions needed for applying an emulator as an proposal density are discussed in [43]. For this purpose, we need to be able to: **(a)** update the construction of the emulator, **(b)** evaluate the emulator, **(c)** normalize the function defined by the emulator, and **(d)** draw samples from the emulator. It is not straightforward to find an interpolative construction which satisfies all those conditions jointly, for an arbitrary dimension of the problem. However, the resulting algorithms (when they can be applied) provide good performance, confirming that the interpolative approach deserves more attention.

4.1.4. Contributions

In this work, we leverage the advances in different fields of numerical integration and emulation, in order to design algorithms which build **(a)** better emulators and **(b)** more efficient quadrature rules. The novel algorithms are adaptive schemes which automatically select the nodes of the quadrature and of the resulting emulator. Namely, the set of nodes used by the emulator is sequentially updated by maximizing a suitable acquisition function. Below, we list the main contributions of the work.

• We propose a novel design of a suitable acquisition function defined as product of the posterior and a diversity term, taking into account the current positions of the nodes. Note that, unlike several works in the literature, e.g., [10, 4, 59, 57], we consider jointly both: the information regarding the posterior and the distances among the current nodes. For the selection of the nodes, some authors also consider the use of MCMC runs [77] or more sophisticated procedures combining sampling and deterministic quadrature schemes for selecting the nodes [74]. Unlike [77, 74], our adaptive approach is based on an active learning procedure. We also provide cheap versions of the acquisition function. The cheap acquisition functions do not require the evaluation of the posterior but only the evaluation of the emulator. The overall schemes are then *parsimonious* techniques which require the evaluation of the posterior density only at the nodes, sequentially selected by optimizing a cheap acquisition function. The proposed active learning strategy is also connected to the idea of obtaining a finite set of weighted *representative points* which can summarize, in some sense, a distribution. This topic has gained attention in the last years [14, 37, 42, 44].

• We consider an interpolative approximation of the posterior density $\bar{\pi}(\mathbf{x})$, where the interpolant is expressed as a linear combination of generic kernel-basis functions. Unlike several BQ techniques in [55, 26, 7], we approximate $\bar{\pi}(\mathbf{x})$ instead of the function $f(\mathbf{x})$ in the integral I. For this purpose, we also propose the combination of the interpolant approach with MC and other quadrature schemes.

• With respect to other schemes in the literature [36, 63], our assumptions regarding the kernel-basis functions are less restrictive, e.g., they do not need to be symmetric. We could also employ different type of bases jointly, e.g., one different basis for each node. For instance, our framework allows the use of nearest neighbors (NN) basis functions, which presents several advantages: it does not require any matrix inversion and the coefficients of the linear combination (which defines the interpolator) are always positive [33],

obtaining always a positive estimation of the marginal likelihood. These benefits are very appealing as shown in [55, 26, 35, 33].

• Section 4.5 presents an importance sampling (IS) interpretation of the proposed schemes, where the weights involve the interpolant instead of the true posterior density. This again shows that we can improve the Monte Carlo approximations without requiring additional evaluations of $\bar{\pi}(\mathbf{x})$. Moreover, the alternative IS interpretation allows to design different techniques. One possible example is given in the final part of Section 4.5.

• We also introduce a novel procedure for fitting the bandwidth parameter of the Gaussian kernel in order to build an *emulator of a density function*. In this scenario, the proposed strategy performs better than the standard maximization of the marginal likelihood of the corresponding GP. Using this tuning procedure, we always obtain positive estimation of the marginal likelihood, even with Gaussian kernels (this is an important point; see [55, 26]).

We provide the theoretical support for the proposed methods in Section 4.7. Most of the convergence results are mainly known in the scattered data approximation literature [68, 76, 60]. The efficiency of the proposed schemes is also confirmed by several numerical experiments (in Section 4.8) with different target pdfs and dimensions of the problem. One of them is also a challenging astronomical application, where the goal is to detect the number of exoplanets orbiting a star, and infer their orbital parameters.

4.2. Interpolative quadratures for Bayesian inference

In many signal processing applications, the goal is to infer a variable of interest given a set of observations or measurements. Let us denote the variable of interest by $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$, and let $\mathbf{y} \in \mathbb{R}^{d_y}$ be the observed data. The posterior pdf is then

$$\bar{\pi}(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{Z(\mathbf{y})},$$

where $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf, and $Z(\mathbf{y})$ is the model evidence (a.k.a. marginal likelihood). Generally, $Z(\mathbf{y})$ is unknown, so we are able to evaluate the unnormalized target function,

$$\pi(\mathbf{x}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}).$$

Usually, the analytical computation of the posterior density $\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x})$ is unfeasible, hence numerical approximations are required. Our goal is to approximate integrals of the form

$$I = \int_{\mathcal{X}} f(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x} = \frac{1}{Z}\int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \tag{4.1}$$

where $f(\mathbf{x})$ is some integrable function, and

$$Z = \int_{\mathcal{X}} \pi(\mathbf{x})d\mathbf{x}. \tag{4.2}$$

In the literature, random sampling or deterministic quadratures are often used [46, 12, 65]. In this work, we consider alternative quadrature rules based on an adaptive interpolative procedure. The adaptation is obtained by applying an active learning scheme.

4.2.1. Interpolative approach

Let us consider a set of distinct nodes $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathcal{X}$ and some non-negative kernel or basis function, $k(\mathbf{x}, \mathbf{x}') : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+ \cup \{0\}$ (i.e., $k(\mathbf{x}, \mathbf{x}') \geq 0$). From now on, we use the terms basis or kernel as synonyms. The interpolant of $\pi(\mathbf{x})$ is as follows

$$\widehat{\pi}(\mathbf{x}) = \sum_{i=1}^{N} \beta_i k(\mathbf{x}, \mathbf{x}_i), \tag{4.3}$$

where the coefficients β_i must be such that $\widehat{\pi}(\mathbf{x})$ interpolates the points $\pi(\mathbf{x}_1), \ldots, \pi(\mathbf{x}_N)$, that is, $\widehat{\pi}(\mathbf{x}_i) = \pi(\mathbf{x}_i)$ for $i = 1, \ldots, N$. Hence, the β_i are the solutions to the following linear system

$$\begin{cases} \beta_1 k(\mathbf{x}_1, \mathbf{x}_1) + ... + \beta_N k(\mathbf{x}_1, \mathbf{x}_N) = \pi(\mathbf{x}_1), \\ \beta_1 k(\mathbf{x}_2, \mathbf{x}_1) + ... + \beta_N k(\mathbf{x}_2, \mathbf{x}_N) = \pi(\mathbf{x}_2), \\ \qquad \vdots \\ \beta_1 k(\mathbf{x}_N, \mathbf{x}_1) + ... + \beta_N k(\mathbf{x}_N, \mathbf{x}_N) = \pi(\mathbf{x}_N). \end{cases} \tag{4.4}$$

Denoting $(\mathbf{K})_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ $(1 \leq i, j \leq N)$, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_N]^\top$ and $\mathbf{d} = [\pi(\mathbf{x}_1), \ldots, \pi(\mathbf{x}_N)]^\top$, Eq. (4.4) can be written in matrix form as $\mathbf{K}\boldsymbol{\beta} = \mathbf{d}$. Thus, the coefficients are given by

$$\boldsymbol{\beta} = \mathbf{K}^{-1}\mathbf{d}. \tag{4.5}$$

Note that, depending on the choice of kernel and its parameters, these coefficients can be negative.

Remark 1. *The only requirement regarding the functions $k(\boldsymbol{x}, \boldsymbol{x}')$ is that the interpolation matrix $\mathbf{K}$ must be non-singular (i.e., invertible) for any set of distinct nodes. The symmetry of $k(\boldsymbol{x}, \boldsymbol{x}')$ is not required. Different type of bases can be employed, for instance, one for each node $\mathbf{x}_i$, i.e., $k_i(\boldsymbol{x}, \boldsymbol{x}_i)$.*

Remark 2. *For simplicity, in this first part of the paper, we consider a fixed number of nodes N. However, a key point of the work is the adaptation procedure in Section 4.6, where new nodes are sequentially added.*

A detailed theoretical analysis is provided in Section 4.7.

4.2.2. Interpolative quadrature schemes

We can approximate both Z and I by substituting the true $\pi(\mathbf{x})$ with its interpolant $\widehat{\pi}(\mathbf{x})$.
Approximation of Z. Let $\int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}_i) d\mathbf{x} = C_i > 0$ be the measure of the i-th kernel. An

approximation of Z can be obtained, by substituting Eq. (4.3) in (4.2),

$$\widehat{Z} = \int_{\mathcal{X}} \widehat{\pi}(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{N} \beta_i \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}_i) d\mathbf{x} = \sum_{i=1}^{N} \beta_i C_i. \tag{4.6}$$

If the kernels are normalized, i.e., $C_i = 1$, note that $\widehat{Z} = \sum_{i=1}^{N} \beta_i$.

Remark 3. *Although $Z > 0$, $\widehat{Z}$ can take negative values, since the coefficients β_i can be negative. However, in this work, we suggest two schemes (with Gaussian bases and a suitable tuning procedure, and with NN bases) which ensure a positive estimation of Z.*

Approximation of I. By substituting (4.3) and (4.6) in (4.1), we obtain an approximation of I as

$$I \approx \widehat{I} = \frac{1}{\widehat{Z}} \int_{\mathcal{X}} f(\mathbf{x}) \widehat{\pi}(\mathbf{x}) d\mathbf{x}. \tag{4.7}$$

Note that, given $\widehat{\pi}(\mathbf{x}) = \sum_{i=1}^{N} \beta_i k(\mathbf{x}, \mathbf{x}_i)$, the approximation of I in (4.7) can be expressed as

$$\begin{aligned} \widehat{I} = \frac{1}{\widehat{Z}} \sum_{i=1}^{N} \beta_i \int_{\mathcal{X}} f(\mathbf{x}) k(\mathbf{x}, \mathbf{x}_i) d\mathbf{x} &= \frac{1}{\widehat{Z}} \sum_{i=1}^{N} \beta_i J_i, \\ &= \frac{1}{\widehat{Z}} \sum_{i=1}^{N} \nu_i \pi(\mathbf{x}_i), \end{aligned} \tag{4.8}$$

where $J_i = \int_{\mathcal{X}} f(\mathbf{x}) k(\mathbf{x}, \mathbf{x}_i) d\mathbf{x}$, $\boldsymbol{\nu} = [\nu_1, ..., \nu_N]^\top = \mathbf{K}^{-1} \boldsymbol{\zeta}$ with $\boldsymbol{\zeta} = [J_1, \ldots, J_N]^\top$ being the vector of integrals. Clearly, the performance of $\widehat{I}$ depends on the discrepancy between $\widehat{\pi}(\mathbf{x})$ and $\pi(\mathbf{x})$, as shown by Theorem 1. This discrepancy is reduced by properly adding new nodes, as suggested in Section 4.6.

4.2.3. Monte-Carlo based interpolative quadrature schemes

In this work, we assume that the evaluation of the target function $\pi(\mathbf{x})$ is the main computational bottleneck [10, 72]. We consider that other operations, such as sampling and evaluating different proposal densities, are negligible with respect to the target evaluation. The techniques, presented in this section, do not require additional target evaluations with respect to Eq. (4.8). In some specific cases, we can compute the integrals J_i and C_i analytically (e.g., see next section). Otherwise, we need to approximate J_i, and in some cases, also C_i. Some general ideas are described below.

Normalized kernels ($C_i = 1$). If the values $C_i = 1$ are known,[3] we can compute $\widehat{Z} = \frac{1}{N} \sum_{n=1}^{N} \beta_i$. Moreover, if we are able to draw samples from each $k(\mathbf{x}, \mathbf{x}_i)$, we have

$$J_i = \int_{\mathcal{X}} f(\mathbf{x}) k(\mathbf{x}, \mathbf{x}_i) d\mathbf{x} \approx \widehat{J_i} = \frac{1}{M} \sum_{m=1}^{M} f(\mathbf{z}_{i,m}), \tag{4.9}$$

[3] For the sake of simplicity and without loss of generality, we assume $C_i = 1$.

with $\mathbf{z}_{i,m} \sim k(\mathbf{x}, \mathbf{x}_i)$, hence

$$\widehat{I} \approx \frac{1}{\widehat{Z}M} \sum_{i=1}^{N} \beta_i \sum_{m=1}^{M} f(\mathbf{z}_{i,m}). \tag{4.10}$$

If we know C_i, another possible scenario is when we are not able to draw from $k(\mathbf{x}, \mathbf{x}_i)$. In this case, we can employ the importance sampling (IS) procedure described below to approximate the integrals J_i.

Kernels with unknown C_i. In this case, we also have to approximate $\int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}_i) d\mathbf{x} = C_i$. For this purpose, we can employ IS with proposal densities $q_i(\mathbf{x})$, with $i = 1, ..., N$, obtaining

$$C_i \approx \widehat{C}_i = \frac{1}{M} \sum_{m=1}^{M} w_{i,m}, \tag{4.11}$$

where the weights are $w_{i,m} = \frac{k(\mathbf{z}_{i,m}, \mathbf{x}_i)}{q_i(\mathbf{z}_{i,m})}$ and $\mathbf{z}_{i,m} \sim q_i(\mathbf{x})$. Moreover, we also obtain

$$J_i \approx \widehat{J}_i = \frac{1}{M} \sum_{m=1}^{M} w_{i,m} f(\mathbf{z}_{i,m}). \tag{4.12}$$

Replacing (4.11)-(4.12) into (4.8), the final estimator is given by

$$\widehat{I} \approx \frac{1}{\sum_{i=1}^{N} \beta_i \sum_{m=1}^{M} w_{i,m}} \sum_{i=1}^{N} \beta_i \sum_{m=1}^{M} w_{i,m} f(\mathbf{z}_{i,m}), \tag{4.13}$$

$$= \sum_{m=1}^{M} \sum_{i=1}^{N} \bar{\rho}_{i,m} f(\mathbf{z}_{i,m}), \tag{4.14}$$

where $\bar{\rho}_{i,m} = \frac{\beta_i w_{i,m}}{\sum_{j=1}^{N} \sum_{k=1}^{M} \beta_j w_{j,k}}$.

Remark 4. *Note that, in any of the scenarios above, we do not need to evaluate the target $\pi(\boldsymbol{x})$ at the samples $\boldsymbol{z}_{i,m}$. Namely, we do not require additional target evaluations with respect to Section 4.2.2. Moreover, as $M \to \infty$, the estimators in Eqs.* (4.10)-(4.14) *converge to the expression* (4.8)*, under standard MC arguments [65].*

For further details, see the theoretical results in Section 4.7.2 and Theorems 6 and 7. So far we have considered Monte Carlo approaches to estimate J_i and C_i. Other particular and more efficient approaches (such as deterministic quadratures) are possible if we consider specific kernel functions. In the next sections, we analyze two specific cases (with Gaussian and NN kernels).

4.3. Interpolation with Gaussian kernels

Let us consider the case of Gaussian kernels (with an unbounded support $\mathcal{X} = \mathbb{R}^{d_x}$),

$$k_G(\mathbf{x}, \mathbf{x}_i) = \frac{1}{(2\pi)^{\frac{d_x}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \mathbf{x}_i)\right), \tag{4.15}$$

where $\boldsymbol{\Sigma}$ is a positive definite matrix. We take $\boldsymbol{\Sigma} = h^2\mathbf{I}$ where $h > 0$ is the bandwidth hyperparameter that needs to be tuned (see Section 4.3.1). Alternatively, note that we can also use unnormalized Gaussian kernels $k_G(\mathbf{x}, \mathbf{x}_i) = A \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \mathbf{x}_i)\right)$, where A is another parameter to possibly tune, and then consider $C_i = A(2\pi)^{\frac{d_x}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}$.

Polynomial functions $f(\mathbf{x})$. The use of Gaussian kernel functions $k_G(\mathbf{x}, \mathbf{x}_i)$ with $f(\mathbf{x})$ being polynomial, ensures that the integrals in (4.8) are available in closed-form. Let $\mathbf{f}(\mathbf{x}) = \mathbf{x}^r = [x_1^r, \ldots, x_{d_x}^r]^\top$ be componentwise powers of $\mathbf{x} \in \mathbb{R}^{d_x}$ ($r = 1, 2, \ldots$). Then,

$$J_i = \int_{\mathbb{R}^{d_x}} \mathbf{f}(\mathbf{x}) k_G(\mathbf{x}, \mathbf{x}_i) d\mathbf{x} = \int_{\mathbb{R}^{d_x}} \mathbf{x}^r k_{\mathrm{G}}(\mathbf{x}, \mathbf{x}_i) d\mathbf{x},$$

corresponds to the r-th marginal moments of a multivariate Gaussian centered at $\mathbf{x}_i$. Note that the marginal moments of a Gaussian density are well-known. Some instances are

$$\int_{\mathbb{R}^{d_x}} \mathbf{x} k_{\mathrm{G}}(\mathbf{x}, \mathbf{x}_i) d\mathbf{x} = \mathbf{x}_i \qquad (r = 1),$$
$$\int_{\mathbb{R}^{d_x}} \mathbf{x}^2 k_{\mathrm{G}}(\mathbf{x}, \mathbf{x}_i) d\mathbf{x} = \mathbf{x}_i^2 + \mathrm{diag}(\boldsymbol{\Sigma}), \qquad (r = 2),$$

where the power $\mathbf{x}_i^2$ is considered a componentwise operation. Then, in this case, we can directly replace the values of J_i in Eq. (4.8).

Generic functions $f(\mathbf{x})$. Each of the N integrals on the right hand of (4.8) may be also approximated efficiently with a *Gauss-Hermite quadrature* (GH) [38, 29], i.e.,

$$\int_{\mathbb{R}^{d_x}} f(\mathbf{x}) k_G(\mathbf{x}, \mathbf{x}_i) d\mathbf{x} \approx \widehat{J_i} = \sum_{m=1}^{M} \bar{w}_m^{\mathrm{GH}} f(\mathbf{z}_{i,m}),$$

where $\bar{w}_m^{\mathrm{GH}}$ and $\mathbf{z}_{i,m}$ are the weights and nodes of the GH quadrature used for i-th integral. Note the quadrature weights are independent of i and are normalized, i.e., $\sum_{m=1}^{M} \bar{w}_m^{\mathrm{GH}} = 1$. Moreover, we have $\mathbf{z}_{i,m} = \widetilde{\mathbf{z}}_m + \mathbf{x}_i$, that is, the only difference is a translation of a single set of GH nodes $\widetilde{\mathbf{z}}_m$ [29] (see also the `Suppl. Material`). Again, we do not need extra evaluations of the target $\pi(\mathbf{x})$. Note that, with enough number of points $\mathbf{z}_{i,m}$, Gauss-Hermite quadrature is also exact when $f(\mathbf{x})$ are polynomial functions [19]. Theoretical results, valid for positive definite radial basis functions, can be found in Section 4.7.2.

4.3.1. Probabilistic interpretation

If $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$ (i.e., it is symmetric) and $k(\mathbf{x}, \mathbf{x}')$ is semi positive definite, as in the Gaussian case, we can interpret the construction of the interpolant $\widehat{\pi}(\mathbf{x})$ as a Gaussian process (GP) [64]. In our setting, $\mathbf{d} = [\pi(\mathbf{x}_1), \ldots, \pi(\mathbf{x}_N)]^\top$ represents the observed vector. The process starts by placing a GP prior on $\pi(\mathbf{x})$, $\pi(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}'))$, where the GP mean is $\mathbf{0}$ and $k(\mathbf{x}, \mathbf{x}')$ is the covariance function. Conditioning on $\mathbf{d}$, it can be shown that the posterior of $\pi(\mathbf{x})$ is given by

$$\pi(\mathbf{x})|\mathbf{d} \sim \mathcal{GP}(\widehat{\pi}(\mathbf{x}), C(\mathbf{x}, \mathbf{x}')),$$

where the mean function is the interpolant $\widehat{\pi}(\mathbf{x})$ given in (4.3), and the posterior covariance function is $C(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x})^\top \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}')$, with

$$\mathbf{k}(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_N)]^\top,$$

and $(\mathbf{K})_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. The variance at $\mathbf{x}$ is

$$V(\mathbf{x}) = C(\mathbf{x}, \mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^\top \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}). \tag{4.16}$$

Observe that $V(\mathbf{x}_i) = 0$ for all $i = 1, \ldots, N$. If we assume that the vector of evaluations $\mathbf{d}$ is noisy, we can relax the exact fit requirement by introducing a regularization term, replacing $\mathbf{K}$ with the matrix $\mathbf{K} + \sigma^2 \mathbf{I}$, where $\mathbf{I}$ is an $N \times N$ identity matrix. The noise term σ^2 also provides numerical stability. The probabilistic interpretation of the integrals involving π is given in Appendix 4.10.2.

4.3.2. Tuning of hyperparameters

Let us denote as $\boldsymbol{\theta}$ the vector as hyperparameters of the kernel functions $k(\mathbf{x}, \mathbf{x}')$. A standard way of fitting the hyperparameters $\boldsymbol{\theta}$ is to maximize the marginal likelihood of the GP [64]. In this case, the evaluations of $\pi(\mathbf{x})$ play the role of data. Given the evaluations $\mathbf{d} = [\pi(\mathbf{x}_1), \ldots, \pi(\mathbf{x}_N)]^\top$, the marginal likelihood is given by $p(\mathbf{d}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{d}|\mathbf{0}, \mathbf{K})$, and its log-version is

$$\log p(\mathbf{d}|\boldsymbol{\theta}) = -\frac{1}{2}\mathbf{d}^\top \mathbf{K}^{-1} \mathbf{d} - \frac{1}{2} \log |\mathbf{K}| + c,$$

where c is a constant. Note that $\mathbf{K}$ depends on $\boldsymbol{\theta}$. However, for fitting the bandwidth parameter h of the Gaussian kernels, we propose an alternative procedure described in Appendix 4.10.1, specifically designed for building an emulator of a *density function*. In this context, the proposed procedure performs better then the maximization of $p(\mathbf{d}|\boldsymbol{\theta})$.

Remark 5. *Using the novel tuning procedure in Appendix 4.10.1, the corresponding estimator $\widehat{Z}$ takes always positive values.*

4.4. Constant kernels based on Nearest Neighbors

Given the set of nodes $\{\mathbf{x}_i\}_{i=1}^N$ in a bounded domain $\mathcal{X}$, consider now the use of constant kernels with finite support

$$k(\mathbf{x}, \mathbf{x}_i) = \mathbb{I}_{\mathcal{R}_i}(\mathbf{x}), \tag{4.17}$$

where $\mathbb{I}_{\mathcal{R}_i}(\mathbf{x})$ is the indicator function in $\mathcal{R}_i$, i.e., $\mathbb{I}_{\mathcal{R}_i}(\mathbf{x}) = 1$ for all $x \in \mathcal{R}_i$ and zero otherwise. Each $\mathcal{R}_i$ consists of the points $\mathbf{x} \in \mathcal{X}$ that are closest to $\mathbf{x}_i$, i.e.,

$$\mathcal{R}_i = \{\mathbf{x} \in \mathcal{X} : \|\mathbf{x} - \mathbf{x}_i\|_p \leq \min_{j \neq i} \left\|\mathbf{x} - \mathbf{x}_j\right\|_p\},$$

where $\|\cdot\|_p$ denotes the p-norm. That is, $\mathcal{X} = \cup_{i=1}^N \mathcal{R}_i$ is the Voronoi partition of $\mathcal{X}$ using $\{\mathbf{x}_i\}_{i=1}^N$ as support points. In this case, solving (4.5) for the coefficients $\boldsymbol{\beta}$ is straightforward since the matrix $\mathbf{K}$ is the identity matrix, and thus

$$\beta_i = \pi(\mathbf{x}_i) \text{ for } i = 1, \ldots, N.$$

Note that all $\beta_i \geq 0$ with this kernel. Hence the interpolant is given by

$$\widehat{\pi}(\mathbf{x}) = \sum_{i=1}^N \pi(\mathbf{x}_i)\mathbb{I}_{\mathcal{R}_i}(\mathbf{x}). \tag{4.18}$$

Note that to evaluate $\widehat{\pi}(\mathbf{x})$ at any $\mathbf{x}$ we need to find just the closest node. We do not need to know the borders of regions $\{\mathcal{R}_i\}_{i=1}^N$ for this purpose. This choice of kernels has three clear advantages:

(i) no need to solve the linear system in (4.5) since $\mathbf{K} = \mathbf{I}$ and hence $\boldsymbol{\beta} = \mathbf{d}$,

(ii) the coefficients $\boldsymbol{\beta} = \mathbf{d}$ are always non-negative (this ensures that $\widehat{Z} \geq 0$),

(iii) no need of tuning the bandwidth hyperparameter.

The difficulty, however, is determining the Voronoi partition, as well as the measures $C_i = \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}_i)d\mathbf{x}$. We show how to address these issues in Section 4.4.1. In this case,

$$C_i = \int_{\mathcal{X}} \mathbb{I}_{\mathcal{R}_i}(\mathbf{x})d\mathbf{x} = |\mathcal{R}_i|,$$

where $|\mathcal{R}_i|$ denotes the measure of the i-th Voronoi region. The approximation of Z is given by

$$\widehat{Z} = \sum_{i=1}^N \pi(\mathbf{x}_i)C_i, \tag{4.19}$$

and Eq. (4.8) is expressed as

$$\begin{aligned}\widehat{I} &= \frac{1}{\widehat{Z}} \sum_{i=1}^N \pi(\mathbf{x}_i) \int_{\mathcal{R}_i} f(\mathbf{x})d\mathbf{x}, \\ &= \frac{1}{\sum_{k=1}^N \pi(\mathbf{x}_k)C_k} \sum_{i=1}^N \pi(\mathbf{x}_i) \int_{\mathcal{R}_i} f(\mathbf{x})d\mathbf{x}. \end{aligned} \tag{4.20}$$

The convergence of this scheme is guaranteed as N grows, as shown by Theorems 8 and 9. Further theoretical analysis are provided in Section 4.7.3. Note that we need to estimate the measures C_i, as well as the integrals $\int_{\mathcal{R}_i} f(\mathbf{x})d\mathbf{x}$ to compute $\widehat{Z}$ and $\widehat{I}$. The next section is devoted to this purpose.

4.4.1. Approximating Voronoi regions and resulting estimators

In order to approximate C_i, we can generate M uniform vectors $\{\mathbf{z}_m\}_{m=1}^M$ in $\mathcal{X}$ via Monte Carlo sampling or Quasi-Monte Carlo sequences (e.g. a Sobol sequence) [12]. Define the set $\mathcal{U}_i$ as

$$\begin{aligned}\mathcal{U}_i &= \{\mathbf{z}_m : \|\mathbf{z}_m - \mathbf{x}_i\|_p \leq \min_{j\neq i} \left\|\mathbf{z}_m - \mathbf{x}_j\right\|_p\} \\ &= \{\mathbf{z}_{\ell_i}\}_{\ell_i=1}^{|\mathcal{U}_i|},\end{aligned}$$

i.e., the $|\mathcal{U}_i|$ vectors closest to $\mathbf{x}_i$ in p-norm, which form a discrete approximation of $\mathcal{R}_i$. Note that $\sum_{i=1}^N |\mathcal{U}_i| = M$. Hence, the measure C_i can be approximated by noting that $\frac{C_i}{|\mathcal{X}|} \approx \frac{|\mathcal{U}_i|}{M}$, hence

$$C_i \approx \frac{|\mathcal{U}_i|}{M}|\mathcal{X}|, \tag{4.21}$$

where $|\mathcal{X}|$ is the measure of $\mathcal{X}$. Thus, the estimator in Eq. (4.19) can be rewritten as

$$\widehat{Z} \approx \frac{|\mathcal{X}|}{M}\sum_{i=1}^N \pi(\mathbf{x}_i)|\mathcal{U}_i|. \tag{4.22}$$

We can also obtain an approximation of the integral $J_i = \int_{\mathcal{R}_i} f(\mathbf{x})d\mathbf{x}$ by leveraging a QMC or MC approximation of the Voronoi regions. Specifically, the uniform vectors $\mathbf{z}_{\ell_i}$ in $\mathcal{U}_i$ can be used to approximate the integral in (4.20) as follows

$$J_i = \int_{\mathcal{R}_i} f(\mathbf{x})d\mathbf{x} \approx \frac{C_i}{|\mathcal{U}_i|}\sum_{\ell_i=1}^{|\mathcal{U}_i|} f(\mathbf{z}_{\ell_i}) \approx \frac{|\mathcal{X}|}{M}\sum_{\ell_i=1}^{|\mathcal{U}_i|} f(\mathbf{z}_{\ell_i}), \tag{4.23}$$

where we used (4.21) again in (4.23). The procedure above can be seen as an accept-reject method, and the estimators are also unbiased [46, Chapter 3 and Section 6.6]. Note that a simpler possible approximation with one point is $J_i = \int_{\mathcal{R}_i} f(\mathbf{x})d\mathbf{x} \approx f(\mathbf{x}_i)C_i$. Thus, replacing the expressions (4.22)-(4.23) in (4.20), the final estimator becomes

$$\widehat{I} \approx \frac{1}{\sum_{k=1}^N \pi(\mathbf{x}_k)|\mathcal{U}_k|}\sum_{i=1}^N \pi(\mathbf{x}_i)\sum_{\ell_i=1}^{|\mathcal{U}_i|} f(\mathbf{z}_{\ell_i}). \tag{4.24}$$

Connection with Section 4.2.3. The estimators above can be interpreted as the application of an importance sampling (IS) scheme as described in Section 4.2.3, for kernel functions with unknown C_i. However, unlike in Section 4.2.3, here we consider a unique and uniform proposal density

$$q_i(\mathbf{x}) = q(\mathbf{x}) = \frac{1}{|\mathcal{X}|}\mathbb{I}_{\mathcal{X}}(\mathbf{x}), \qquad \forall i = 1, ..., N.$$

Then, we can also remove the subindex i in the sample $\mathbf{z}_{i,m} \sim q(\mathbf{x})$, i.e., we have only M samples $\mathbf{z}_m \sim q(\mathbf{x})$. Hence, following Eqs. (4.11)-(4.12), we have

$$C_i \approx \frac{1}{M}\sum_{m=1}^M w_{i,m}, \tag{4.25}$$

$$J_i = \int_{\mathcal{R}_i} f(\mathbf{x})d\mathbf{x} \approx \frac{1}{M}\sum_{m=1}^M w_{i,m} f(\mathbf{z}_m), \tag{4.26}$$

where $\mathbf{z}_m \sim q(\mathbf{x}) = \frac{1}{|\mathcal{X}|}\mathbb{I}_{\mathcal{X}}(\mathbf{x})$, and the weights are

$$w_{i,m} = \frac{k(\mathbf{z}_m, \mathbf{x}_i)}{q(\mathbf{z}_m)} = \begin{cases} |\mathcal{X}| & \text{if} \quad \mathbf{z}_m \in \mathcal{R}_i, \\ 0 & \text{if} \quad \mathbf{z}_m \notin \mathcal{R}_i. \end{cases} \tag{4.27}$$

Replacing the expression of the weights $w_{i,m}$ into the formulas above, we recover the estimators in (4.22) and (4.24).

4.5. An alternative IS interpretation

In this section, we discuss a special case of the IS scheme given in Section 4.2.3, when a unique proposal $q_i(\mathbf{x}) = q(\mathbf{x})$ is employed and only M samples $\mathbf{z}_m \sim q(\mathbf{x})$ are drawn (as already considered in the previous section). In this scenario, the IS procedure in Section 4.2.3 has another relevant interpretation, which allows us to design other different schemes. Considering a generic kernel $k(\mathbf{x}, \mathbf{x}_i)$ and Eq. (4.25), we can rearrange $\widehat{Z}$ as

$$\begin{aligned} \widehat{Z} = \sum_{i=1}^{N} \beta_i C_i &\approx \sum_{i=1}^{N} \beta_i \frac{1}{M} \sum_{m=1}^{M} w_{i,m} \\ &= \sum_{i=1}^{N} \beta_i \frac{1}{M} \sum_{m=1}^{M} \frac{k(\mathbf{z}_m, \mathbf{x}_i)}{q(\mathbf{z}_m)} \\ &= \frac{1}{M} \sum_{m=1}^{M} \frac{\sum_{i=1}^{N} \beta_i k(\mathbf{z}_m, \mathbf{x}_i)}{q(\mathbf{z}_m)}. \end{aligned}$$

Then, recalling that $\widehat{\pi}(\mathbf{x}) = \sum_{i=1}^{N} \beta_i k(\mathbf{x}, \mathbf{x}_i)$ and replacing this expression above, we finally obtain

$$\widehat{Z} \approx \frac{1}{M} \sum_{m=1}^{M} \frac{\widehat{\pi}(\mathbf{z}_m)}{q(\mathbf{z}_m)} = \frac{1}{M} \sum_{m=1}^{M} \gamma_m, \tag{4.28}$$

where $\gamma_m = \gamma(\mathbf{z}_m) = \frac{\widehat{\pi}(\mathbf{z}_m)}{q(\mathbf{z}_m)}$ for $m = 1, ..., M$. Moreover, with similar steps, we can obtain

$$\widehat{I} \approx \frac{1}{M\widehat{Z}} \sum_{m=1}^{M} \gamma_m f(\mathbf{z}_m), \tag{4.29}$$

Remark 6. *The weights γ_m have the form of the standard IS weights with the target function $\widehat{\pi}$ in the numerator, and the proposal density q in the denominator. Hence, the entire sampling procedure can be interpreted as a standard IS scheme where the target function is $\widehat{\pi}$ instead of π. This shows again that we do not need extra target evaluations and, hence, we can employ an arbitrary large value of M.*

Remark 7. *Note that this result is valid for any kernel $k(\boldsymbol{x}, \boldsymbol{x}_i)$, and we use a unique proposal $q(\boldsymbol{x})$ in the procedure described in Section 4.2.3.*

Below, we consider the NN case with a uniform proposal $q(\mathbf{x})$, deriving the same formulas in Section 4.4.1.

Uniform proposal density and NN interpolator. Let us consider $q(\mathbf{x}) = \frac{1}{|\mathcal{X}|}\mathbb{I}_{\mathcal{X}}(\mathbf{x})$, i.e., a uniform density in $\mathcal{X}$, and the NN kernel function. For each sample $\mathbf{z}_m$, the corresponding weight γ_m is

$$\gamma_m = \gamma(\mathbf{z}_m) = \frac{\widehat{\pi}(\mathbf{z}_m)}{\frac{1}{|\mathcal{X}|}} = \frac{\pi(\mathbf{x}_{k_m})}{\frac{1}{|\mathcal{X}|}} = |\mathcal{X}|\pi(\mathbf{x}_{k_m}),$$

where $\mathbf{x}_{k_m}$ is the closest node to sample $\mathbf{z}_m$, i.e., $\mathbf{x}_{k_m} = \arg\min_j \left\|\mathbf{z}_m - \mathbf{x}_j\right\|_p$. Then, the IS approximation of $\widehat{Z}$ is

$$\widehat{Z} \approx \frac{1}{M}\sum_{m=1}^{M}\gamma_m = \frac{|\mathcal{X}|}{M}\sum_{m=1}^{M}\pi(\mathbf{x}_{k_m}) = \frac{|\mathcal{X}|}{M}\sum_{k=1}^{N}\pi(\mathbf{x}_k)|\mathcal{U}_k|,$$

where $|\mathcal{U}_k|$ counts the number of $\mathbf{z}_m$ whose closest node is $\mathbf{x}_k$ ($k = 1, \ldots, N$). Note that this expression is the same as in (4.22). Similarly, the IS estimate of $\widehat{I}$ is given by

$$\begin{aligned}\widehat{I} \approx \frac{1}{M\widehat{Z}}\sum_{m=1}^{M}\gamma_m f(\mathbf{z}_m) &= \frac{|\mathcal{X}|}{M\widehat{Z}}\sum_{m=1}^{M}\pi(\mathbf{x}_{k_m})f(\mathbf{z}_m) \\ &= \frac{|\mathcal{X}|}{M\widehat{Z}}\sum_{k=1}^{N}\pi(\mathbf{x}_k)\sum_{\ell_k=1}^{|\mathcal{U}_k|}f(\mathbf{z}_{\ell_k}),\end{aligned}$$

which is the same expression as in (4.24). However, this alternative IS interpretation allows us to design different schemes using a different proposal density, as shown below.

Gaussian mixture proposal. We consider now an alternative to the uniform proposal in $\mathcal{X}$. More specifically, we propose drawing $\{\mathbf{z}_\ell\}_{m=1}^{M}$ from a Gaussian mixture proposal pdf built considering the set of nodes $\{\mathbf{x}_i\}_{i=1}^{N}$, i.e.,

$$\mathbf{z}_m \sim q(\mathbf{x}) = \sum_{i=1}^{N}\xi_i\mathcal{N}(\mathbf{x}|\mathbf{x}_i, \mathbf{C}_i),$$

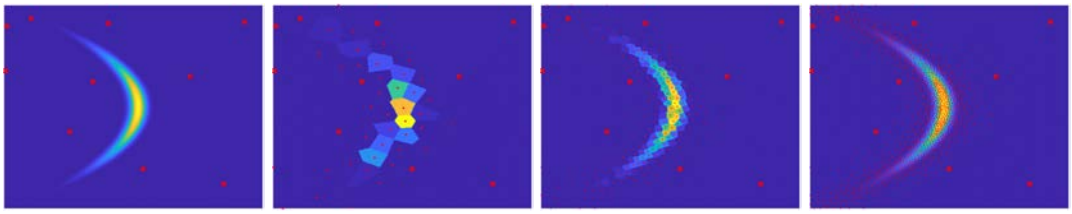
where the mixture weights ξ_i are

$$\xi_i = \frac{\pi(\mathbf{x}_i)}{\sum_{n=1}^{N}\pi(\mathbf{x}_n)}, \qquad i = 1, ..., N,$$

and the covariances $\mathbf{C}_i$ can be determined by the minimum distance of $\mathbf{x}_i$ to its closest node. In this case, the IS weights are given by

$$\gamma(\mathbf{z}_m) = \frac{\widehat{\pi}(\mathbf{z}_m)}{\sum_{i=1}^{N}\xi_i\mathcal{N}(\mathbf{z}_m|\mathbf{x}_i, \mathbf{C}_i)} = \frac{\pi(\mathbf{x}_{k_m})}{\sum_{i=1}^{N}\xi_i\mathcal{N}(\mathbf{z}_m|\mathbf{x}_i, \mathbf{C}_i)},$$

where $\mathbf{x}_{k_m}$ is the closest node to $\mathbf{z}_m$, with $m = 1, \ldots, M$.



(a) True target pdf (b) $E = 50$ (c) $E = 250$ (d) $E = 10^3$

Figure 4.1: Example of application of NN-AQ. The cross-marks represent the starting nodes, while the points added adaptively by NN-AQ are shown with dots. **(a)** The banana-shaped target and the starting nodes. **(b)**-**(c)**-**(d)** The NN-AQ emulator with $E = 50, 250, 10^3$ number of target evaluations.

4.6. Adaptive procedure

In this section, we present an adaptive mechanism to add new nodes to the interpolant. Our algorithm adds nodes sequentially with the aim to discover high-valued regions of $\pi(\mathbf{x})$ while fostering the exploration of the state space. We employ an active learning procedure where a new point is obtained by maximizing a suitable acquisition function. The resulting adaptive algorithm is shown in Table 4.1. Note that the final number of nodes is $N_T = T + N_0$. The adaptive quadrature scheme based on the Gaussian kernels is denoted as GK-AQ, whereas the other scheme based on the Nearest Neighbors (NN) kernels is denoted as NN-AQ. Figure 4.1 depicts an example of application of the NN-AQ.

4.6.1. Building suitable acquisition functions

Let us denote as $t \in \mathbb{N}$ the tth iteration of the algorithm. In the update stage, we decide to add a new node where the acquisition function, $A_t : \mathcal{X} \to \{0\} \cup \mathbb{R}^+$, is maximum. The acquisition function takes into account the shape of $\pi(\mathbf{x})$ and the spatial distribution of the current nodes. More specifically, it must fulfill

$$A_t(\mathbf{x}_i) = 0 \quad \text{for all } t \text{ and } \quad i = 1, \dots, N_t,$$

and grow as we move apart from the nodes. We consider acquisition functions $A_t(\mathbf{x})$ of the form

$$A_t(\mathbf{x}) = \pi(\mathbf{x}) D_t(\mathbf{x}), \tag{4.30}$$

where $D_t(\mathbf{x})$ is a diversity term that penalizes the proximity to the current nodes. Note that the information of $f(\mathbf{x})$ could be also included as $A_t(\mathbf{x}) = f(\mathbf{x})\pi(\mathbf{x})D_t(\mathbf{x})$. In some settings, the function $A_t(\mathbf{x})$ above could be directly used after choosing a diversity term $D_t(\mathbf{x})$. However, in this work, we consider that evaluating $\pi(\mathbf{x})$ is costly, so we propose cheaper versions of (4.30).

Table 4.1: **Adaptive Quadrature algorithm.**

Initialization: Set N_0 initial nodes and set $\mathbf{X}_0 = \{\mathbf{x}_1, \dots, \mathbf{x}_{N_0}\}$, $\mathbf{d}_0 = [\pi(\mathbf{x}_1), \dots, \pi(\mathbf{x}_{N_0})]^\top$.

For $t = 0, \dots, T$**:**

1. *Build the interpolator*. Use the set $\mathbf{X}_t = \{\mathbf{x}_1, \dots, \mathbf{x}_{N_t}\}$ and corresponding evaluations $\mathbf{d}_t = [\pi(\mathbf{x}_1), \dots, \pi(\mathbf{x}_{N_t})]^\top$ to build $\widehat{\pi}_t(\mathbf{x})$ using Gaussian kernels (see Section 4.3) or constant kernels (see Section 4.4).

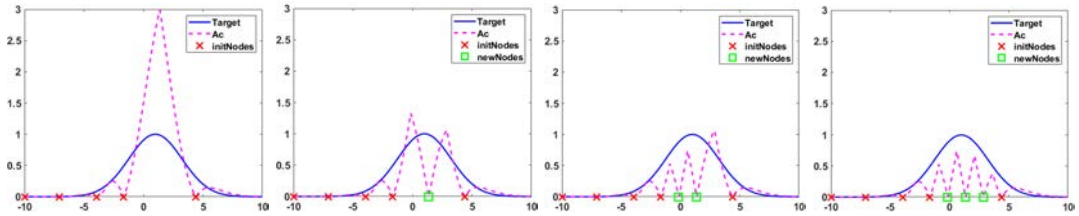
2. *Build the acquisition function*. Use $\widehat{\pi}_t(\mathbf{x})$ and the set of current nodes $\mathbf{X}_t$ to build the acquisition function $A_t(\mathbf{x})$, e.g., Eqs. (4.33)-(4.34).

3. *Update stage*. Obtain new node $\mathbf{x}_{N_t+1}$ by

$$\mathbf{x}_{N_t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} A_t(\mathbf{x}), \tag{4.31}$$

   append $\mathbf{X}_{t+1} = \{\mathbf{X}_t, \mathbf{x}_{N_t+1}\}$ and $\mathbf{d}_{t+1} = [\mathbf{d}_t, \pi(\mathbf{x}_{N_t+1})]^\top$.

Outputs: Build the final interpolant $\widehat{\pi}_T(\mathbf{x})$ and obtain the approximations $\widehat{I}$ and $\widehat{Z}$.



(a) Initial state (b) Iteration 1 (c) Iteration 2 (d) Iteration 3

Figure 4.2: 1D example of application of $A_t(\mathbf{x}) = \pi(\mathbf{x})D_t(\mathbf{x})$ with the diversity term $D_t(\mathbf{x}) = \min_{i=1,\dots,N_t} |x - x_i|$. At each iteration, the new node, shown with a green square, is added where $A_t(x)$ is maximum.

4.6.2. Cheap acquisition functions

We recall that the most costly step is the evaluation of the target function $\pi(\mathbf{x})$. This is often due to the use of complex models and/or large amounts of data. For that reason, we propose a cheap type of $A_t(\mathbf{x})$,

$$A_t(\mathbf{x}) = \widehat{\pi}_t(\mathbf{x})D_t(\mathbf{x}), \tag{4.32}$$

so that no evaluations of the true $\pi(\mathbf{x})$ are required. In this case, in terms of posterior evaluations E, the cost of the overall algorithm in Table 4.1, is $E = N_0 + T$.

Remark 8. *The particular case $A_t(\boldsymbol{x}) = D_t(\boldsymbol{x})$ corresponds to the space-filling experi-*

mental designs (e.g., see [59, 60, 51] and Theorem 4). In the other particular case with $A_t(\boldsymbol{x}) = \widehat{\pi}_t(\boldsymbol{x})$, the resulting schemes are similar to other approaches in literature which combine sampling and optimization (e.g., see [2]).

In the Gaussian kernel scenario, we may use the variance in (4.16) as diversity term

$$A_t(\mathbf{x}) = \widehat{\pi}_t(\mathbf{x}) V_t(\mathbf{x}), \tag{4.33}$$

where we have set $D_t(\mathbf{x}) = V_t(\mathbf{x})$, that fulfills $V_t(\mathbf{x}_i) = 0$ for $i = 1, \dots, N_t$. This choice is motivated by the fact that the approximation error is bounded by the maximum value of $V_t(\mathbf{x})$ (e.g., see Theorem 3). Since the function $V_t(\mathbf{x})$ is unfeasible with constant NN kernels, we suggest a diversity term of the form

$$A_t(\mathbf{x}) = \widehat{\pi}_t(\mathbf{x}) \min_{i=1,\dots,N_t} \|\mathbf{x} - \mathbf{x}_i\|_p \,. \tag{4.34}$$

Note that the term $D_t(\mathbf{x}) = \min_{i=1,\dots,N_t} \|\mathbf{x} - \mathbf{x}_i\|_p$ is zero when evaluated at any current node: for each $\mathbf{x}_j \in \mathbf{X}_t$ the minimum distance is w.r.t. itself, which is zero. This choice is motivated by Theorem 4, since the approximation error is also bounded by the maximum value of $D_t(\mathbf{x})$. Figure 4.2 depicts an example with this choice of $D_t(\mathbf{x})$. Note that the choice $D_t(\mathbf{x}) = \min_{i=1,\dots,N_t} \|\mathbf{x} - \mathbf{x}_i\|_p$ can be also employed in the Gaussian kernel scenario.

Another alternative is to consider tempering versions of the acquisition function,

$$A_t(\mathbf{x}) = \left[\widehat{\pi}_t(\mathbf{x})\right]^{\alpha} \left[D_t(\mathbf{x})\right]^{\beta}, \tag{4.35}$$

where $\alpha \geq 0$ can be used to prioritize moving towards high-valued zones of $\widehat{\pi}_t(\mathbf{x})$, while $\beta \geq 0$ to encourage exploration. The values α and β can also vary with the iteration t. The maximization of $A_t(\mathbf{x})$ can be performed by simulated annealing or other optimization techniques. The performance of different acquisition functions have been compared in Figure 4.4 (see Section 4.8.1). One can observe that maximizing the proposed acquisition functions provides much better results than adding uniformly random nodes.

Observations. For the GK-AQ algorithm, the most costly step corresponds to the inversion of the $N_t \times N_t$ matrix $\mathbf{K}_t$, needed to be done in order to build the acquisition function in Eq. (4.33). Note that the inverse $\mathbf{K}_t^{-1}$ is used for both evaluating the interpolant $\widehat{\pi}_t(\mathbf{x})$ and computing the variance $V_t(\mathbf{x})$. We can alleviate the cost of this step by building $\mathbf{K}_t^{-1}$ iteratively from $\mathbf{K}_{t-1}^{-1}$. The recursion formula is given in Appendix 4.10.3. In the case of NN-AQ, evaluating the acquisition function in (4.34) requires only to calculate the distances with respect to each node. This computation can be used for both evaluating the interpolant and the diversity term $D_t(\mathbf{x}) = \min \|\mathbf{x} - \mathbf{x}_i\|_p$. Note that the cost of searching for the nearest neighbor has only a weak dependence on the dimension of the space.

4.7. Theoretical support

In this section, we provide some theoretical results supporting the proposed schemes. We consider $\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$ a bounded target pdf and a bounded domain $\mathcal{X} \subset \mathbb{R}^{d_x}$. Let also

$f(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$ be an integrable function. In this section, we consider $J = \int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$ as the integral of interest. For a generic $f(\mathbf{x})$, J corresponds to the numerator of the integral I in Eq. (4.1). For $f(\mathbf{x}) = 1$, J becomes the normalizing constant of $\pi(\mathbf{x})$, i.e., $J = Z$, which is the denominator of I. Thus, working with J is equivalent to working with I. Let also $\widetilde{J} = \int_{\mathcal{X}} f(\mathbf{x})\widehat{\pi}(\mathbf{x})d\mathbf{x}$, be the approximation of J given by substituting the interpolant $\widehat{\pi}(\mathbf{x})$. A first general result valid for any interpolation procedure is given below.

Theorem 1. *The error incurred by substituting $\pi(\boldsymbol{x})$ with $\widehat{\pi}(\boldsymbol{x})$ in J is bounded,*

$$\begin{aligned} |J - \widetilde{J}| &\leq \left\| f(\pi - \widehat{\pi}) \right\|_1 \\ &\leq \|f\|_2 \left\| \pi - \widehat{\pi} \right\|_2 \\ &\leq |\mathcal{X}| \, \|f\|_\infty \left\| \pi - \widehat{\pi} \right\|_\infty , \end{aligned}$$

where $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$ denote the L^1, L^2 and L^∞ norms respectively.

Proof. See Appendix 4.10.4. □

Therefore, if are able to build an interpolant $\widehat{\pi}$ in a way such $\left\| \pi - \widehat{\pi} \right\|_\infty$ vanishes to zero, then the approximation $\widetilde{J}$ will converge to J. Note that, in this section, we ensure the convergence of numerator J and denominator Z of $I = \frac{J}{Z}$, independently. A complete treatment (yet more complicated) should consider the convergence of the two quantities at the same time. For the rest of results, we need to distinguish between the case of Gaussian kernel and constant kernel interpolators. To establish convergence of both schemes we need to make some preliminary definitions and considerations.

4.7.1. Space-filling measures and related results

We introduce two well-known measures of dispersion widely employed in the function approximation literature. In this section, we always consider a bounded support $\mathcal{X}$.
Fill distance. Given the set of nodes $\{\mathbf{x}_i\}_{i=1}^N \subset \mathcal{X}$, let us define the following quantity

$$r = \max_{\mathbf{x} \in \mathcal{X}} \min_{1 \leq i \leq N} \|\mathbf{x} - \mathbf{x}_i\|_2 , \tag{4.36}$$

which is the fill distance.
Separation distance. The separation distance is defined as

$$s = \min_{i \neq j} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2 , \tag{4.37}$$

i.e., the minimal distance between two nodes. Note that $s \leq 2r$. Having a small s increases the numerical instability and can have a detrimental effect in the error bounds. The adaptive procedure described in Sect. 4.6 produces a sequence of nodes that sequentially minimizes r.

Proposition 1. *Consider the acquisition function given in Eq.* (4.35) *with $\alpha = 0$ and $\beta = 1$, and the choice $A_t(\boldsymbol{x}) = \min_{i=1,\dots,N_t} \|\boldsymbol{x} - \boldsymbol{x}_i\|_2$, where $\{\boldsymbol{x}_i\}_{i=1}^{N_t}$ are the current nodes of the*

interpolator. The maximum of this function is the fill distance r_t in Eq. (4.36)*, at iteration t. Adding the point $\boldsymbol{x}_{N_t+1}$ corresponding to r_t to the set of current nodes ensures that*

$$r_{t+1} = \max_{i=1,\dots,N_{t+1}} \min \|\boldsymbol{x} - \boldsymbol{x}_i\|_2 \leq r_t,$$

and that $r_t \to 0$ when $t \to \infty$.

Proof. See Sect. 4.1 in [60] and [4]. This procedure is related to the "coffee house design" in [51]. □

Proposition 2. *For isotropic kernels, the variance function $V(\boldsymbol{x})$ given in Eq.* (4.16) *satisfies that $\max_{\boldsymbol{x}\in\mathcal{X}}[V(\boldsymbol{x})]^{\frac{1}{2}} \leq \Phi(r)$, where $\Phi(r)$ is an increasing function of r, depending on the kernel function. In the case of Gaussian kernels, $\Phi(r)$ is an exponential function.*

Proof. See Sect. 2.1 in [60] and Sect. 2 in [4]. □

Proposition 3. *Consider the acquisition function given in Eq.* (4.35) *with $\alpha = 0$ and $\beta = 1$, i.e., and the choice $A_t(\boldsymbol{x}) = V_t(\boldsymbol{x})$. Let us set also $\varphi_t = \max_{\boldsymbol{x}\in\mathcal{X}} V_t(\boldsymbol{x})$. By adding new nodes according to the rule*

$$\boldsymbol{x}_{N_t+1} = \arg\max A_t(\boldsymbol{x}),$$

we are minimizing φ_t over the iterations t, i.e., φ_t is a non-increasing function of t and $\varphi_t \to 0$ as $t \to \infty$.

Proof. This algorithm is known as p-greedy algorithm in [66]. See the behavior of the variance of a GP interpolant [64]. This acquisition function is commonly used in the kriging literature. For instance, see [41] and [59]. □

Proposition 4. *Consider the acquisition function given in Eq.* (4.34) *with $\alpha = 0$ and $\beta = 1$, and the choice $A_t(\boldsymbol{x}) = \min_{i=1,\dots,N_t} \|\boldsymbol{x} - \boldsymbol{x}_i\|_2$, where $\{\boldsymbol{x}_i\}_{i=1}^{N_t}$ are the current nodes of the interpolator. The sequence of nodes obtained as $\boldsymbol{x}_{N_t+1} = \arg\max A_t(\boldsymbol{x})$, for $t \in \mathbb{N}^+$, is a uniform low-discrepancy sequence in a bounded $\mathcal{X}$ [53].*

Proof. This procedure can be interpreted as deterministic and sequential version of the well-known latin hypercube sampling (LHS) [53]. □

Remark 9. *Note that the proposed schemes do not need that the space is covered uniformly. The only requirement, for decreasing the fill distance r, is to be able to reach any subset of the domain $\mathcal{X}$ with a non-null probability (strictly positive).*

4.7.2. Results for interpolators based on radial basis functions (RBFs)

In this section, we consider that $k(\mathbf{x}, \mathbf{x}')$ is the Gaussian kernel considered in Sect. 4.3. More generally, the results from this section are valid for any $k(\mathbf{x}, \mathbf{x}')$ that is a (positive definite) radial basis function (RBF).

Exact computation of J_i

Recall $\widehat{\pi}(\mathbf{x}) = \sum_{i=1}^{N} \beta_i k(\mathbf{x}, \mathbf{x}_i)$, where the weights are $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_N] = \mathbf{K}^{-1}\mathbf{d}$ using the interpolation matrix $\mathbf{K}$ and the vector of target evaluations $\mathbf{d}$. The approximation $\widetilde{J}$ can be written as

$$\widetilde{J} = \int_{\mathcal{X}} f(\mathbf{x})\widehat{\pi}(\mathbf{x})d\mathbf{x} = \sum_{i=1}^{N} \beta_i J_i = \sum_{i=1}^{N} \nu_i \pi(\mathbf{x}_i),$$

where $J_i = \int_{\mathcal{X}} f(\mathbf{x})k(\mathbf{x}, \mathbf{x}_i)d\mathbf{x}$, and the weights $\boldsymbol{\nu} = [\nu_1, \ldots, \nu_N]^\top$ are given by $\boldsymbol{\nu} = \mathbf{K}^{-1}\boldsymbol{\zeta}$ with $\boldsymbol{\zeta}$ being the vector of J_i's. In this form, $\widetilde{J}$ is expressed as a combination of evaluations of $\pi(\mathbf{x})$, i.e., a quadrature. The following theorem establishes that the weights $\boldsymbol{\nu} = \mathbf{K}^{-1}\boldsymbol{\zeta}$ are optimal for a quadrature of this kind. Note that the Gaussian kernels are symmetric positive definite functions, and are special cases of radial basis functions (RBF).

Theorem 2. *Let us consider a symmetric kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_j, \boldsymbol{x}_i)$ which always defines a positive definite matrix $\mathbf{K}$. The native space related to $k(\boldsymbol{x}, \boldsymbol{x}')$ is a reproducing kernel Hilbert space (RKHS) [3, 69]. Given the points $\{\boldsymbol{x}_i\}_{i=1}^N$ and $\boldsymbol{\nu} = \mathbf{K}^{-1}\boldsymbol{\zeta}$, the quadrature $\widetilde{J} = \sum_{i=1}^{N} \nu_i \pi(\boldsymbol{x}_i)$ is optimal in the sense of Golomb-Weinberg [23], i.e., the weights ν_i minimizes the norm of the integration error functional in the dual space [3, 69].*

Proof. A sketch of the proof is in App. 4.10.4. See also [70] and [7] and references therein. □

Theorem 3. *Suppose that $\pi(\boldsymbol{x})$ belongs to the RKHS generated by the kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$. The interpolant $\widehat{\pi}(\boldsymbol{x}) = \sum_{i=1}^{N} \beta_i k(\boldsymbol{x}, \boldsymbol{x}_i)$ satisfies $|\pi(\boldsymbol{x}) - \widehat{\pi}(\boldsymbol{x})| \leq \|\pi\|_{\mathcal{H}} [V(\boldsymbol{x})]^{\frac{1}{2}}$ for all $\boldsymbol{x} \in \mathcal{X}$ and hence $\left\|\pi - \widehat{\pi}\right\|_\infty \leq \|\pi\|_{\mathcal{H}} \max_{\boldsymbol{x}\in\mathcal{X}} [V(\boldsymbol{x})]^{\frac{1}{2}}$, where $\|\cdot\|_{\mathcal{H}}$ denotes the norm in the RKHS, and $V(\boldsymbol{x})$ is the variance function given in Eq.* (4.16). *Hence, from Theorem 1, we have*

$$|J - \widetilde{J}| \leq |\mathcal{X}| \, \|f\|_\infty \, \|\pi\|_{\mathcal{H}} \max_{\boldsymbol{x}\in\mathcal{X}} [V(\boldsymbol{x})]^{\frac{1}{2}}.$$

Proof. See Sect. 2.1 in [60] and Sect. 2 in [4]. □

The theorem above, jointly with Proposition 3, justify the choice of the diversity term $D_t(\mathbf{x}) = V_t(\mathbf{x})$ in Section 4.6.2. The next theorem, based on results from the literature on approximating functions with RBFs, establishes that the approximation error tends to zero when $r \to 0$, and that the rate of convergence can be exponentially fast in the case of infinitely smooth RBFs, such as the Gaussian kernels.

Theorem 4. *The error of the quadrature $\widetilde{J}$ is*

$$|J - \widetilde{J}| \leq |\mathcal{X}| \, \|f\|_\infty \left\|\pi - \widehat{\pi}\right\|_\infty = O(\lambda(r)),$$

where $\lambda(r) \to 0$ as $r \to 0$, with r being the fill distance given in Eq. (4.36). *The convergence rate depends on the regularity degree of $\pi(\boldsymbol{x})$. For $\pi(\boldsymbol{x})$ sufficiently regular (technically, belonging to the RKHS induced by the RBF kernel), and Gaussian RBF the bound $\lambda(r)$ decreases exponentially*

$$\lambda(r) = e^{-c_h |\log r|/r},$$

with a certain constant $c_h > 0$, which generally depends on the bandwidth h.

Proof. See Sect. 11.3 and table in page 188 of [76]. □

Recall that the diversity term in (4.34) produces a monotonically decreasing sequence of fill distances that converges to zero in the limit of $t \to \infty$, as stated in Proposition 1. The next theorem states that the approximation error tends to zero as $N \to \infty$, and provides a quite pessimistic upper bound.

Theorem 5. *Given a sequence of nodes $\{\boldsymbol{x}_i\}_{i=1}^N$ generated as in Proposition 4, it can be shown that $r \leq C_{d_x,\mathcal{X}} N^{-1/d_x} \log N$, where $C_{d_x,\mathcal{X}}$ is a constant that probably depends on the dimension d_x and the measure of $\mathcal{X}$. Then, the following (pessimistic) upper bound can be provided*

$$|J - \widetilde{J}| = O\left(e^{-c_1 \frac{1}{N^{-1/d_x} \log N} - c_2 \frac{\left|\log\left(N^{-1/d_x} \log N\right)\right|}{N^{-1/d_x} \log N}}\right),$$

where $c_1 > 0$ and $c_2 > 0$ are constants depending on h, d_x and the measure of $\mathcal{X}$.

Proof. See Sect. 2.5.1 in [60] and [53]. □

Noisy computation of J_i

Theorem 4 above states that the convergence of $\widetilde{J}$ is achieved when the fill distance r goes to zero. Recall that in $\widetilde{J} = \sum_{i=1}^N \beta_i J_i$ we consider the exact computation of $J_i = \int_{\mathcal{X}} f(\mathbf{x}) k(\mathbf{x}, \mathbf{x}_i) d\mathbf{x}$. In this section, we consider of approximating J_i by the estimator $\widehat{J_i}$, so that we finally have a noisy version of $\widetilde{J}$, i.e., $\widehat{J} = \sum_{i=1}^N \beta_i \widehat{J_i}$. Below, we show some results related to $\widehat{J}$, but we need some previous definitions.

Stability. The numerical stability of the solution depends on the inversion of the interpolation matrix $\mathbf{K}$ and it is connected to the separation distance s. Clearly, if two nodes are very close, then the corresponding two rows of the interpolation matrix are almost identical and the matrix becomes ill-conditioned [67, 76].

Reproduction quality. Roughly speaking, an interpolant built with more nodes (i.e., N grows) filling the space, generally yields a better approximation. This concept is connected to the fill distance r in Eq. (4.36). Recall that the fill distance is a measure of how well the data fills the space [76].

Uncertainty principle. A typical problem when reconstructing functions is the trade-off

between reproduction quality and numerical stability. Let us consider RBF kernels with a *fixed bandwidth*, as N grows. Generally, when one aims at a very good approximation of the function of interest, the numerical stability gets compromised, and conversely, if one aims to have good numerical stability, the approximation will be poor. This is known in the literature as uncertainty principle [67].

Let us denote as h the parameter which controls the bandwidth of the RBFs, as $\boldsymbol{\Sigma} = h^2\mathbf{I}$ in the Gaussian kernel. The next theorem illustrates the case where the numerical instability combined with the error in computing the vector of integrals $\boldsymbol{\zeta} = [J_1, \dots, J_N]^\top$ deteriorates the error bound of Theorem 4 (for a fixed h). Let us denote the vector of approximated integrals by $\widehat{\boldsymbol{\zeta}} = [\widehat{J_1}, \dots, \widehat{J_N}]^\top$ and recall $\mathbf{d} = [\pi(\mathbf{x}_1), \dots, \pi(\mathbf{x}_N)]^\top$ is the vector of evaluations of π.

Theorem 6. (for a fixed bandwidth h**)** *Let us consider a bounded support $\mathcal{X}$. If we take into account the error in the evaluation of the integrals $\boldsymbol{\zeta} = [J_1, \dots, J_N]^\top$, denoted by $\widehat{\boldsymbol{\zeta}} = [\widehat{J_1}, \dots, \widehat{J_N}]^\top$, the corresponding approximation $\widehat{J} = \sum_{i=1}^N \beta_i \widehat{J_i}$ has an error of*

$$\begin{aligned} |J - \widehat{J}| &\leq |\mathcal{X}|\, \|f\|_\infty \left\|\pi - \widehat{\pi}\right\|_\infty + \|\mathbf{K}^{-1}\|_2 \|\mathbf{d}\|_2 \|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\|_2 \\ &= O(\lambda(r)) + O(\upsilon(s,h))\|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\|_2, \end{aligned}$$

where $\lambda(r) \to 0$ as $r \to 0$, $\upsilon(s,h) \to \infty$ as $s \to 0$, with r and s being, respectively, the fill distance and separation distance given in Eqs. (4.36) *and* (4.37)*. The parameter h, which determines the bandwidth of the radial kernel, is considered fixed. The function $\upsilon(s,h)$ is an upper bound for $\left\|\mathbf{K}^{-1}\right\|_2$, which is a measure of stability (note that $\|\mathbf{K}^{-1}\|_2$ corresponds to the inverse of the lowest eigenvalue of $\mathbf{K}$).*

Proof. See Appendix 4.10.4. For the bound $\upsilon(s,h)$ see Corollary 12.4 in [76]. □

The bound in Theorem 6 expresses the uncertainty relation. Indeed, we see that making $s \to 0$ poses a problem if we use a fixed bandwidth h. Indeed, the interpolation matrix $\mathbf{K}$ becomes ill-conditioned as two nodes are too close, and the error $\|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\|_2$ is amplified. The growing rate of $\upsilon(s,h)$, as $\lambda(r)$, depends on the smoothness of the RBF. For Gaussian kernels, the rates of $\upsilon(s,h)$ and $\lambda(r)$ are both exponential. However, with a Monte Carlo approximation, we can always improve the approximation $\widehat{\boldsymbol{\zeta}}$ by increasing the number of samples M, so that $\|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\|_2 \to 0$. Recall that the increase of the number of Monte Carlo samples M does not require additional evaluations of the target π in the proposed schemes. Furthermore, even with a fixed M, we can control the value $\|\mathbf{K}^{-1}\|_2$ by decreasing the bandwidth h of the kernel function. The following results consider these two cases.

Theorem 7. (for a fixed bandwidth h **and** $M \to \infty$**)** *Given a bounded support $\mathcal{X}$, consider the application of a Monte Carlo method to approximate $\boldsymbol{\zeta}$, then $\|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\|_2 \to 0$ as $M \to \infty$, where M is the number of samples. Hence, the approximation $\widehat{J} = \sum_{i=1}^N \beta_i \widehat{J_i}$ has an error*

$$|J - \widehat{J}| = O(\lambda(r)),$$

where $\lambda(r) \to 0$ *as the fill distance* $r \to 0$ *and* $M \to \infty$.

Proof. The term $\|\boldsymbol{\zeta} - \widetilde{\boldsymbol{\zeta}}\|_2 \to 0$ as the number of Monte Carlo samples $M \to \infty$ [65]. □

Conjecture 1. (for a decreasing bandwidth h **and fixed** M**)** *Given a bounded support* $\mathcal{X}$*, consider a noisy approximation* $\widehat{\boldsymbol{\zeta}}$ *of* $\boldsymbol{\zeta}$*. Assume that we decrease* h *as the number of nodes* N *grows (in order to control the instability term, i.e., the magnitude of* $\left\|\mathbf{K}^{-1}\right\|_2$*). Hence, the approximation* $\widehat{J} = \sum_{i=1}^{N} \beta_i \widehat{J}_i$ *has an error*

$$|J - \widehat{J}| = O(\lambda(r)) + b,$$

where b *is some constant bias,* $\lambda(r) \to 0$ *as* $r \to 0$*, and making* $h \to 0$ *when* $N \to \infty$.

Note that, as h approaches 0, the interpolation matrix $\mathbf{K}$ becomes a diagonal matrix, with the maximum values of the kernels in the diagonal. Thus, controlling the maximum values of the kernel functions, we can control the minimum value of the eigenvalues, such that the interpolation matrix $\mathbf{K}$ be well-conditioned. Moreover, recall that we are using an interpolative approach and the probabilistic interpretation in Section 4.3.1 is not strictly required. Therefore, we have more flexibility in the choice and/or tuning of the kernel functions. Indeed, one could consider different bandwidths (one for each kernel function), bigger in regions with lower density of points, while smaller bandwidths in regions with a higher density of nodes. This would improve the numerical stability.

Remark 10. *The interplant based on NN kernels does not suffer the uncertainty problem, since they have compact non-overlapping supports. Namely, we can interpret that the bandwidths are automatically tuned.*

4.7.3. Results for local interpolators

In a local interpolation method, the addition and/or a change of one node, only affects the solution in a subset of the support domain. This scenario corresponds to the use of the constant NN kernels. Recall that the interpolant based on constant kernels,

$$\widehat{\pi}(\mathbf{x}) = \sum_{i=1}^{N} \pi(\mathbf{x}_i) \mathbb{I}_{\mathcal{R}_i}(\mathbf{x}),$$

where $\mathcal{R}_i$ denotes the Voronoi region associated with node $\mathbf{x}_i$. Let us first state a result for sufficiently smooth $\pi(\mathbf{x})$. If $\pi(\mathbf{x})$ is Lipschitz continuous, i.e., for all $\mathbf{x}, \mathbf{z} \in \mathcal{X}$ we have $|\pi(\mathbf{z}) - \pi(\mathbf{x})| \leq L_0 \|\mathbf{z} - \mathbf{x}\|$ for some constant L_0, then we have the following result.

Theorem 8. *Given the NN interpolant* $\widehat{\pi}(\boldsymbol{x})$*, if* $\pi(\boldsymbol{x})$ *is Lipschitz continuous we have that* $\left\|\pi - \widehat{\pi}\right\|_\infty \leq L_0 r$*, where* L_0 *is the Lipschitz constant and* r *is the fill distance introduced in Eq.* (4.36)*. Then, from Theorem 1, we have*

$$|J - \widetilde{J}| \leq |\mathcal{X}| \, \|f\|_\infty \, L_0 r.$$

Moreover, given a sequence of nodes $\{\boldsymbol{x}_i\}_{i=1}^N$ *generated as in Proposition 4, and since* $r \leq C_{d_x,\mathcal{X}} N^{-1/d_x} \log N$*, we have the following (pessimistic) bound*

$$|J - \widetilde{J}| = \mathcal{O}\left(N^{1/d_x} \log N\right).$$

Proof. See Appendix 4.10.4. □

Now, recall the approximation of $\widetilde{J}$ given by

$$\widetilde{J} = \int_{\mathcal{X}} f(\mathbf{x})\widehat{\pi}(\mathbf{x})d\mathbf{x} \approx S_N = \sum_{i=1}^{N} \pi(\mathbf{x}_i) f(\mathbf{x}_i) C_i,$$

where S_N is the Riemann approximation, which has been also discussed in Sect. 4.4.1, and $C_i = \int_{\mathcal{R}_i} d\mathbf{x}$, i.e., the measure of $\mathcal{R}_i$. Here, we used the approximation $\int_{\mathcal{R}_i} f(\mathbf{x})d\mathbf{x} \approx f(\mathbf{x}_i)C_i$. We will show that S_N converges to $J = \int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$ as we add more nodes according to one of the proposed acquisition functions, that is, as $t \to \infty$. As with Gaussian kernels, the convergence is related with how well the nodes fill space. Here, the role of fill distance is played by the maximum of the measures C_i. The theorem below states that, as we fill the space, the measures C_i converges to zero. Recall that the Voronoi partition $\{\mathcal{R}_i\}_{i=1}^N$ generated from the set of nodes $\{\mathbf{x}_i\}_{i=1}^N$ corresponds to the subdivision of $\mathcal{X}$ in N non-overlapping pieces.

Proposition 5. *Consider a sequence of points* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ *covering the space* $\mathcal{X}$*, then for the associated Voronoi regions* $\mathcal{R}_i$*, we have that* $\max_i C_i \to 0$ *as* $N \to \infty$*.*

Proof. See the proofs of Theorems 1 and 4 in [17]. □

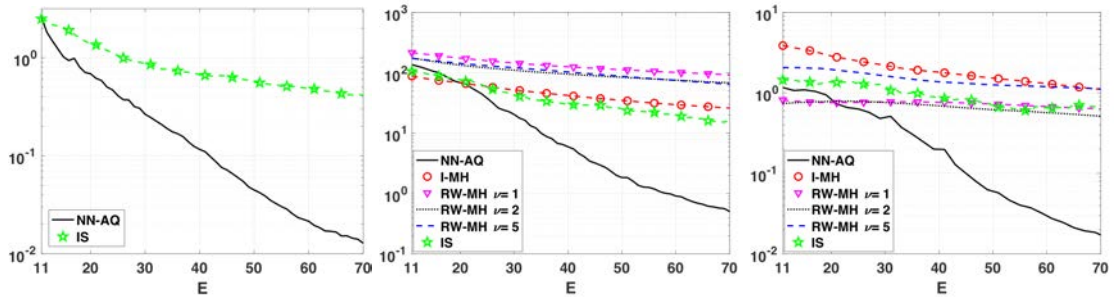
Theorem 9. *Let* $\pi(\boldsymbol{x})$ *be a continuous and bounded target pdf (up to a normalizing constant) defined on a bounded support* $\mathcal{X} \subset \mathbb{R}^{d_x}$*. Let* $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ *bounded on* $\mathcal{X}$*. Consider the integral* $J = \int_{\mathcal{X}} f(\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x}$*. Let us consider a Voronoi partition of* $\mathcal{X}$*, generated by the nodes* $\{\boldsymbol{x}_i\}_{i=1}^N$*, defined as* $\mathcal{R}_1, \ldots, \mathcal{R}_N$ *(recall that* $C_i = |\mathcal{R}_i|$*). Given the Riemann sum* $S_N = \sum_{i=1}^N f(\boldsymbol{x}_i)\pi(\boldsymbol{x}_i)C_i$*, the convergence of* $S_N \to J$ *is guaranteed as* $\max_i C_i \to 0$ *when* $N \to \infty$*.*

Proof. See Sect 8.3 in [61]. □

Above, we have assumed that C_i are known. However, we can have very accurate Monte Carlo estimates without requiring additional evaluations of the target $\pi(\mathbf{x})$ (but just of the interpolant $\widehat{\pi}(\mathbf{x})$), i.e., only with a slight increase in the overall computation cost.

4.8. Numerical experiments

In this section, we provide several numerical tests in order to show the performance of the proposed adaptive quadrature schemes and compare them with benchmark approaches



(a) (b) (c)

Figure 4.3: **(a)** Rel-MSE in log-scale for Z as function of number of target evaluations E. **(b)** Rel-MSE in log-scale for $\boldsymbol{\mu}$ as function of number of target evaluations E. **(c)** Rel-MSE in log-scale for estimating $[\sigma_1^2, \sigma_2^2]$ as function of number of target evaluations E.

in the literature. The first example corresponds to a nonlinear banana-shaped density in dimension d_x =2, 3, 4 and 5. The second test is a multimodal scenario with dimension d_x=10. Finally, we test our schemes in a challenging astronomic inference problem of detecting the number of exoplanets orbiting a star.

4.8.1. Banana target

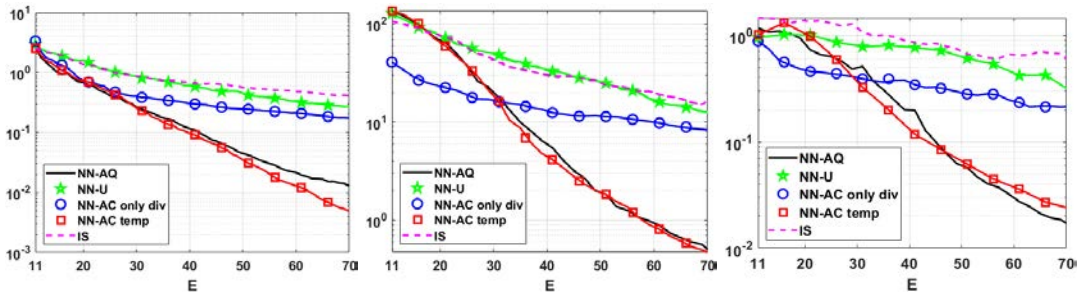
As a first example, we consider a banana-shaped target pdf,

$$\bar{\pi}(\mathbf{x}) \propto \exp\left\{-\frac{(\eta_1 - Bx_1 - x_2^2)^2}{2\eta_0^2} - \sum_{i=1}^{d_x} \frac{x_i^2}{2\eta_i^2}\right\}, \tag{4.38}$$

with $\mathbf{x} \in \mathcal{X} = [-10, 10]^{d_x}$, $B = 4$, $\eta_0 = 4$ and $\eta_i = 3.5$ for $i = 1, ..., d_x$. We consider $d_x = \{2, 3, 4, 5\}$ (i.e., different dimensions) and compute in advance the *true* moments of the target (i.e., the groundtruth) by using a costly grid, in order to check the performance of the different techniques.

Experiment 1

We set $d_x = 2$ and test the different algorithms in order to compute the vector mean $\boldsymbol{\mu} = [-0.4, 0]$ and the diagonal of the covariance matrix $[\sigma_1^2, \sigma_2^2] = [1.3813, 8.9081]$. Moreover, our schemes are also able to estimate Z, whose ground-truth is $Z = 7.9979$, thus we also measure the error in this estimation. We compare the performance in terms of Relative Mean Square Error (Rel-MSE), averaged over 500 independent runs, using different methodologies: **(a)** NN-AQ starting with $N_0 = 10$ nodes randomly chosen in $[-10, 10]\times[-10, 10]$ and $M = 10^5$; **(b)** an independent MH algorithm (I-MH) with random initialization in $[-10, 10] \times [-10, 10]$; **(c)** random-walk MH algorithms (RW-MH) with different proposal variance, and random initialization in $[-10, 10]\times[-10, 10]$; **(d)** an IS algorithm. The proposal density for both I-MH and IS is a uniform in $[-10, 10]\times[-10, 10]$, whereas for the RW-MHs is a Gaussian density centered at the current state of the chain



(a) (b) (c)

Figure 4.4: **(a)** Rel-MSE in log-scale for Z as function of number of target evaluations E. **(b)** Rel-MSE in log-scale for $\boldsymbol{\mu}$ as function of number of target evaluations E. **(c)** Rel-MSE in log-scale for estimating $[\sigma_1^2, \sigma_2^2]$ as function of number of target evaluations E.

with covariance matrix $v^2\mathbf{I}$ where $v \in \{1, 2, 5\}$ (so we consider 3 different RW-MHs).
For a fair comparison, we need that all methods have the same number E of target evaluations (fixing $E = 70$). Since NN-AQ, I-MH and RW-MH require one new target evaluation per iteration, we run $T = 70$ iterations for I-MH and RW-MH ($E = T$), and $T - N_0 = 60$ iterations for NN-AQ. In this regard, the IS algorithm use 70 samples drawn from the uniform proposal. Hence, all methods need $T = 70$ target evaluations. The results are given in Figures 4.3(a)-(b). Note that the estimation of Z via MCMC techniques is not straightforward (e.g., see [40]).

Discussion 1. We can observe that NN-AQ outperforms the other methods in terms of Rel-MSE in estimation. Moreover, in Fig. 4.3(a)-(b) we can see that the decrease is much greater, as E grows, than the other methods. Namely, NN-AQ has more benefits with new evaluations of $\pi(\mathbf{x})$.

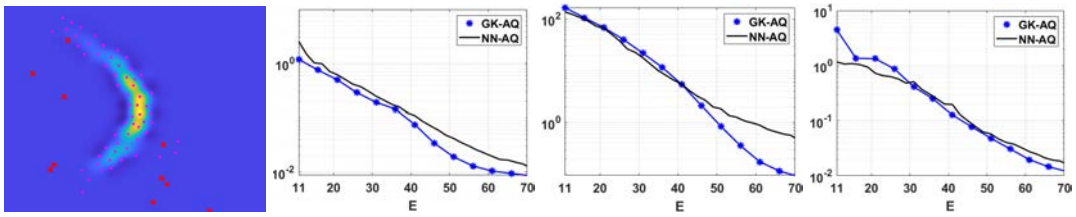
Experiment 2

In this case, we fix the number of target evaluations E, and vary $d_x = \{2, 3, 4, 5\}$. The Rel-MSE in the estimation of Z is given in Table 4.2 (with $E \in \{100, 1000\}$).

Discussion 2. In this experiment, E is fixed along different dimensions. The results given in Table 4.2, with fixed E, does not show all the potential of NN-AQ. However, NN-AQ outperforms IS in all the dimensions d_x considered when $E = 1000$.

Table 4.2: **Relative MSE of Z with $E \in \{100, 1000\}$ for different d_x**

methods	E	$d_x = 2$	$d_x = 3$	$d_x = 4$	$d_x = 5$
NN-AQ	100	0.0027	0.1127	0.3798	1.9730
	1000	$4{\cdot}10^{-4}$	0.0023	0.0140	0.0374
IS	100	0.2645	0.4427	0.7627	1.1115
	1000	0.0226	0.0378	0.0641	0.1094



(a) (b) (c) (d)

Figure 4.5: **(a)** Example of application of GK-AQ with 10 starting points (red cross-marks) and T=60 iterations (red dots), i.e., $E = 70$ target evaluations. **(b)** Rel-MSE in log-scale for Z as function of number of target evaluations E. **(c)** Rel-MSE in log-scale for $\boldsymbol{\mu}$ as function of number of target evaluations E. **(d)** Rel-MSE in log-scale for estimating $[\sigma_1^2, \sigma_2^2]$ as function of number of target evaluations E.

Experiment 3

For $d_x = 2$, we compare now IS, NN-AQ, and three variants of NN-AQ: (i) NN-U, where the optimization step in (4.31) is substituted with sampling uniformly the new node in $[-10, 10] \times [-10, 10]$ (i.e., without using an acquisition function), (ii) NN-AQ *only diversity*, which uses the acquisition in (4.35) with $\alpha = 0, \beta = 1$, i.e., with only the diversity term $D_t(\mathbf{x})$, and (iii) NN-AQ *tempered*, which uses the acquisition in (4.35) with $\alpha = 0$, $\beta_t = \frac{200}{t}$, i.e., $A_t(\mathbf{x}) = [D_t(\mathbf{x})]^{\beta_t}$. Note that the adaptation in NN-AQ *only diversity* can be viewed as filling the space in a deterministic way. Note also that the adaptation in NN-AQ *tempered* will encourage more exploration than NN-AQ in the early iterations. Again, we compare the error in estimating Z, $\boldsymbol{\mu}$ and $[\sigma_1^2, \sigma_2^2]$ as a function of target evaluations E (up to $E = 70$). The results are given in Figures 4.4(a)-(b).

Discussion 3. We can observe that NN-AQ and NN-AQ *tempered* outperform the others in terms of Rel-MSE in estimation. Moreover, in Fig. 4.4(a)-(b) we can see that for NN-AQ and NN-AQ *tempered*, the RMSE decreases at a faster rate as E grows, than the NN-U and NN-AQ *only diversity*, highlighting the importance of taking into account the current interpolant to locate the new nodes. It can be seen that NN-AQ *only diversity* works much better than NN-U in the early iterations. We explain these results by the fact that NN-AQ *only diversity* tends to cover the space more efficiently in these early iterations since it avoids placing new nodes near the existing ones. However, as E grows, the performance of NN-U and NN-AQ *only diversity* is similar since both end up filling uniformly the space. Interestingly, NN-U performs better than IS as E increases, which demonstrate the power of the interpolative approach even when the new nodes are randomly chosen.

Experiment 4

For $d_x = 2$, we investigate the performance of GK-AQ in the estimation of Z, $\boldsymbol{\mu}$ and $[\sigma_1^2, \sigma_2^2]$ as function of E. NN-GK employs the acquisition in (4.33). The kernel bandwidth h is fitted using the procedure in Appendix 4.10.1. As commented in Sect. 4.3.2, we consider a small noise of $\sigma = 10^{-2}$ for numerical stability. We will compare the performance against NN-AQ. The results are given in Figures 4.5(a)-(d), along with an

example of GK-AQ interpolant , with $E = 70$, obtained in a specific run.

Discussion 4. The results are shown in Figures 4.5(b)-(d). GK-AQ outperforms NN-AQ in this particular experiment. However, it is important to remark that the results of GK-AQ may worsen considerably if h is not selected adequately (we have used the procedure in App. 4.10.1), in contrast to NN-AQ which is free of hyperparameter tuning and hence more robust.

4.8.2. Multimodal target

In this experiment, we consider a multimodal Gaussian target in $d_x = 10$,

$$\bar{\pi}(\mathbf{x}) = \frac{1}{3}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \frac{1}{3}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) + \frac{1}{3}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3),$$

with $\boldsymbol{\mu}_1 = [5, 0, \ldots, 0]$, $\boldsymbol{\mu}_2 = [-7, 0, \ldots, 0]$, $\boldsymbol{\mu}_3 = [1, \ldots, 1]$ and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = 4^2\mathbf{I}_{10}$. We want to test the performance of the different methods in estimating the normalizing constant $Z = 1$. We consider an application of GK-AQ with $N_0 = 500$ initial nodes, random in $[-15, 15]^{10}$, and $T = 1000 - N_0$, hence fixing the number of evaluations to $E = 1000$. We compare it against three sophisticated AIS schemes, namely PMC, LAIS and AMIS [8]. For PMC, we choose Gaussian proposal pdfs and test different number of proposals $L \in \{10, 100, 200, 500\}$, whose means are also initialized at random in $[-15, 15]^{10}$. At each iteration one sample is drawn from each proposal, hence the algorithm is run for $T_{\text{PMC}} = \frac{1000}{L}$ iterations for a fair comparison. As a second alternative, we consider the deterministic mixture weighting approach for PMC, which is shown to have better overall performance, denoted DM-PMC. For LAIS, we also consider different number of proposals $L \in \{10, 100, 200, 500\}$. More specifically, we consider two versions of LAIS: the *one-chain* version and an *ideal* version. In ideal LAIS, the means of the L Gaussian proposals are drawn exactly from $\bar{\pi}(\mathbf{x})$. The one-chain application of LAIS (OC-LAIS) requires to run a MCMC algorithm targeting $\bar{\pi}(\mathbf{x})$ to obtain the L proposal means, hence it requires L evaluations of the target. At each iteration one sample is drawn from the mixture of the L Gaussian proposals, hence we run the algorithm for $T_{\text{LAIS}} = 1000 - L$ iterations for a fair comparison. We used a Gaussian random walk Metropolis to obtain the L means in the one-chain scenario. Finally, we consider AMIS with several combinations of number of iterations T_{AMIS} and number of samples per iteration R. At each iteration, R samples are drawn from a single Gaussian proposal, hence the total number of evaluations is $E = RT_{\text{AMIS}}$. In this case, we test $E \in \{1000, 2000, 3000, 5000\}$, so the comparison is not fair except for $E = 1000$. For PMC, LAIS and AMIS, as well as for the random walk proposal within the Metropolis algorithm, the covariance of the Gaussian proposals was fixed to $h^2\mathbf{I}_{10}$ (for $h = 1, ..., 6$), where h is the initial bandwidth parameter used in GK-AQ.[4] All the methods are compared through the mean absolute error (MAE) in estimating Z, and the results are averaged over 500 independent simulations. The results are shown in Table 4.3 and Table 4.4. For each method, the best and worst MAE are boldfaced.

[4]Recall that, for GK-AQ, the final bandwidth is tuned as described in App. 4.10.1.

Discussion. We can observe that GK-AQ obtains the best range of MAE values [0.078, 0.4782] and the best results for $h = 1$. For $h > 1$, we can see in Tables 4.3-4.4 that the lowest MAE values are obtained by ideal LAIS with $L = 500$ and $h = 3$. We stress that ideal LAIS is not available in practice, since we usually cannot sample directly from $\bar{\pi}(\mathbf{x})$. Regardless of the ideal LAIS scheme (not applicable in practice), GK-AQ provides the best results. Moreover, we see that GK-AQ with $h = 3$ is the best performing method in this experiment, since it achieves a lower MAE than PMC, DM-PMC and OC-LAIS for every combination of L and h. Table 4.4 shows that AMIS performs worse than GK-AQ for $E = 1000$ (fair comparison), but even with much more AMIS evaluations $E \in \{2000, 3000\}$ (unfair comparison in favor of AMIS). AMIS needs to reach a big enough value of E ($E = 5000$), to beat GK-AQ in terms of MAE.

Table 4.3: **MAE of Z with $E = 1000$** (best and worst MAE of each method are boldfaced)

Methods		**h = 1**	**h = 2**	**h = 3**	**h = 4**	**h = 5**	**h = 6**
GK-AQ		**0.4782**	0.1741	**0.0780**	0.1362	0.1497	0.2322
PMC	$N = 10$	0.9993	0.9526	0.8603	0.6743	0.6024	0.6155
	$N = 100$	0.9998	0.9896	0.8853	0.6761	0.5192	**0.4544**
	$N = 200$	**1.0002**	0.9893	0.8816	0.7099	0.6389	0.5384
	$N = 500$	0.9995	0.9916	0.9741	0.8700	0.7421	0.6544
DM-PMC	$N = 10$	0.9991	0.9478	0.8505	0.6009	0.5352	0.5814
	$N = 100$	0.9997	0.8719	0.4490	0.2425	**0.1901**	0.2193
	$N = 200$	0.9999	0.9321	0.5708	0.3257	0.2374	0.2524
	$N = 500$	**1.0000**	0.9888	0.7969	0.5009	0.3684	0.3800
Ideal LAIS	$N = 10$	**0.9992**	0.8114	0.2579	0.0863	0.0819	0.1091
	$N = 100$	0.9918	0.3638	0.0547	0.0407	0.0598	0.1053
	$N = 200$	0.9846	0.2486	0.0352	0.0411	0.0680	0.1093
	$N = 500$	0.9687	0.1852	**0.0335**	0.0473	0.0891	0.1353
OC-LAIS	$N = 10$	**1.0000**	1.0000	0.9992	0.9883	0.9468	0.9079
	$N = 100$	0.9999	0.8731	0.4434	0.2785	0.2392	0.2870
	$N = 200$	0.9982	0.7028	0.2418	0.1243	0.1406	0.2070
	$N = 500$	0.9937	0.4949	0.1221	**0.0857**	0.1195	0.1786

4.8.3. Applications to exoplanet detection

In recent years, the problem of revealing objects orbiting other stars has acquired large attention. Different techniques have been proposed to discover exo-objects but, nowadays, the radial velocity technique is still the most used [25, 5, 1, 73]. The problem consists in fitting a dynamical model to data acquired at different moments spanning during long time periods (up to years). The model is highly non-linear and, for certain sets of parameters, its evaluation is quite costly in terms of computation time. This is due to the fact that its

Table 4.4: **MAE of Z of AMIS with $E \in \{1000, 2000, 3000, 5000\}$**

Methods		**h = 1**	**h = 2**	**h = 3**	**h = 4**	**h = 5**	**h = 6**
GK-AQ (E=1000)		**0.4782**	0.1741	**0.0780**	0.1362	0.1497	0.2322
	$M = 10$	0.9998	0.9997	0.9997	0.9996	0.9996	0.9995
AMIS	$M = 100$	1.0000	1.0000	1.0000	0.9999	0.9997	0.9990
$E = 1000$	$M = 200$	1.0000	1.0000	1.0000	1.0000	0.9998	0.9994
	$M = 500$	**1.0000**	1.0000	1.0000	1.0000	0.9998	**0.9989**
	$M = 10$	0.9155	0.9117	0.8981	0.8987	0.8891	**0.8878**
AMIS	$M = 100$	0.9998	0.9986	0.9934	0.9784	0.9559	0.9072
$E = 2000$	$M = 200$	1.0000	1.0000	0.9998	0.9981	0.9888	0.9712
	$M = 500$	**1.0000**	1.0000	1.0000	0.9998	0.9984	0.9953
	$M = 10$	0.3293	0.3402	**0.3051**	0.3381	0.3540	0.3443
AMIS	$M = 100$	0.9725	0.9040	0.7963	0.6384	0.4964	0.3816
$E = 3000$	$M = 200$	0.9998	0.9977	0.9884	0.9527	0.8308	0.7119
	$M = 500$	**1.0000**	1.0000	0.9998	0.9988	0.9859	0.9566
	$M = 10$	0.0766	0.0768	**0.0695**	0.0722	0.0699	0.0725
AMIS	$M = 100$	0.1626	0.1176	0.0957	0.0810	0.0737	0.0656
$E = 5000$	$M = 200$	0.8771	0.6040	0.2824	0.1473	0.1163	0.0899
	$M = 500$	**1.0000**	0.9982	0.9904	0.9449	0.7944	0.4532

evaluation involves numerically integrating a differential equation, or using an iterative procedure for solving a non-linear equation (until a certain condition is satisfied). This loop can be very long for some sets of parameters.

Likelihood function

When analyzing radial velocity data of an exoplanetary system, it is commonly accepted that the *wobbling* of the star around the center of mass is caused by the sum of the gravitational force of each planet independently and that they do not interact with each other. Each planet follows a Keplerian orbit and the radial velocity of the host star is given by

$$y_t = V_0 + \sum_{i=1}^{S} K_i \left[\cos\left(u_{i,t} + \omega_i\right) + e_i \cos\left(\omega_i\right)\right] + \xi_t, \tag{4.39}$$

with $t = 1, \ldots, T$.[5] The number of objects in the system is S. Both y_t, $u_{i,t}$ depend on time t, and ξ_t is a Gaussian noise perturbation with variance σ_e^2. We consider the noise variance σ_e^2 an unknown parameter as well. The meaning of each parameter in Eq. (4.39) is given in Table 4.5. The likelihood function is jointly defined by (4.39) and some indicator variables described below. The angle $u_{i,t}$ is the true anomaly of the planet i and it can be

[5]More generally, we can have y_{t_j} with $j = 1, ..., T$.

Table 4.5: Description of parameters in Eq. (4.39).

Parameter	Description	Units
For each planet		
K_i	amplitude of the curve	$\mathrm{m\,s^{-1}}$
$u_{i,t}$	true anomaly	rad
ω_i	longitude of periastron	rad
e_i	orbit's eccentricity	...
P_i	orbital period	s
τ_i	time of periastron passage	s
Below: not depending on the number of objects/satellite		
V_0	mean radial velocity	$\mathrm{m\,s^{-1}}$

determined from

$$\frac{du_{i,t}}{dt} = \frac{2\pi}{P_i} \frac{(1 + e_i \cos u_{i,t})^2}{(1 - e_i)^{\frac{3}{2}}}$$

This equation has an analytical solution. As a result, the true anomaly $u_{i,t}$ can be determined from the mean anomaly $M_{i,t}$. However, the analytical solution contains a non-linear term that needs to be determined by iterating. First, we define the mean anomaly $M_{i,t}$ as

$$M_{i,t} = \frac{2\pi}{P_i} (t - \tau_i),$$

where τ_i is the time of periastron passage of the planet i and P_i is the period of its orbit (see Table 4.5). Then, through the Kepler's equation,

$$M_{i,t} = E_{i,t} - e_i \sin E_{i,t}, \tag{4.40}$$

where $E_{i,t}$ is the eccentric anomaly. Equation (4.40) has no analytic solution and it must be solved by an iterative procedure. A Newton-Raphson method is typically used to find the roots of this equation [58]. For certain sets of parameters, this iterative procedure can be particularly slow and the computation of the likelihood becomes quite costly. We also have

$$\tan \frac{u_{i,t}}{2} = \sqrt{\frac{1 + e_i}{1 - e_i}} \tan \frac{E_{i,t}}{2}, \tag{4.41}$$

Therefore, the variable of interest $\mathbf{x}$ is the vector of dimension $d_X = 1 + 5S$ (where S is the number of planets),

$$\mathbf{x} = [V_0, K_1, \omega_1, e_1, P_1, \tau_1, \ldots, K_S, \omega_S, e_S, P_S, \tau_S],$$

For a single object (e.g., a planet or a natural satellite), the dimension of $\mathbf{x}$ is $d_X = 5{+}1 = 6$, with two objects the dimension of $\mathbf{x}$ is $d_X = 11$, etc. All the Eqs. from (4.39) to (4.41) induce a likelihood function $\ell(\mathbf{y}|\mathbf{x}, \sigma_e) = \prod_{t=1}^{T} \ell(y_t|\mathbf{x}, \sigma_e)$, where $\mathbf{y} = \{y_1, \ldots, y_T\}$.

Prior and posterior densities

The prior $g(\mathbf{x})$ is defined as multiplication of indicator variables $V_0 \in [-20, 20]$, $K_i \in [0, \max y_{i,t} - \min y_{i,t}]$, $e_i \in [0, 1]$, $P_i \in [0, 365]$, $\omega_{i,t} \in [0, 2\pi]$, $\tau_i \in [0, 30]$, (i.e., the prior is zero outside these intervals), for all $i = 1, \ldots, S$. This means that the prior density is zero when the particles fall out of these intervals. Note that the interval of τ_i is conditioned to the value P_i. This parameter is the time of periastron passage, i.e. the time passed since the object crossed the closest point in its orbit. It has the same units of P_i and can take values from 0 to P_i. The complete posterior is

$$p(\mathbf{x}|\mathbf{y}, \sigma_e) = \frac{1}{p(\mathbf{y}|\sigma_e)} \ell(\mathbf{y}|\mathbf{x}, \sigma_e) g(\mathbf{x}).$$

We are interested in inferring the parameters $\mathbf{x}$ and, more specifically, computing the marginal likelihood

$$Z = p(\mathbf{y}|\sigma_e) = \int_{\mathcal{X}} \ell(\mathbf{y}|\mathbf{x}, \sigma_e) g(\mathbf{x}) d\mathbf{x},$$

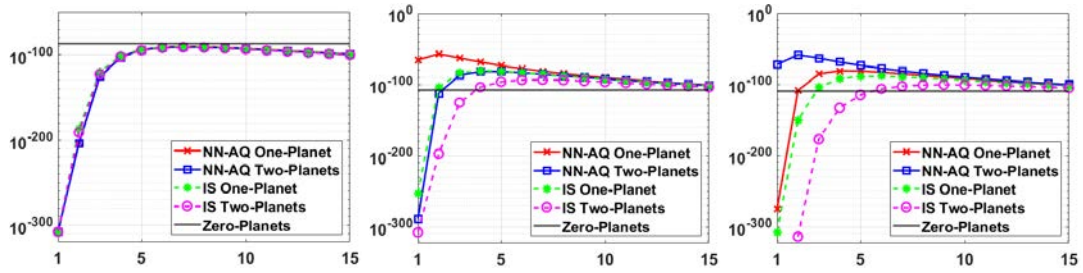
obtained integrating out $\mathbf{x}$, in order to infer the number of planets. The noise variance σ_e^2 is also inferred after the sampling, by maximizing $Z = p(\mathbf{y}|\sigma_e)$, i.e., $\widehat{\sigma_e^2} = \arg\max_{\sigma_e} p(\mathbf{y}|\sigma_e)$.

Experiments

Given a set of data $\mathbf{y}$ generated according to the model (see the initial parameter values below), our goal is to infer the number S of planets in the system. For this purpose, we have to approximate the model evidence $Z = p(\mathbf{y}|\sigma_e)$ of each model. In all experiments, we consider 60 total number of observations. We consider three different experiments: **(E1)** $S = 0$, i.e., no object, **(E2)** $S = 1$ (one object) and **(E2)** the case of two objects $S = 2$. We set $V = 2$, in all cases. For the first object in **E1** and **E2**, we set $K_1 = 25$, $\omega_1 = 0.61$, $e_1 = 0.1$, $P_1 = 15$, $\tau_1 = 3$. For **E2**, we also consider a second object with $K_2 = 5$, $\omega_2 = 0.17$, $e_2 = 0.3$, $P_2 = 115$, $\tau_2 = 25$ (in that case $S = 2$). All the data are generated with $\sigma_e^2 = 2$. The rest of trajectories are generated according to the transition model (and the corresponding measurements y_t according to the observation model).

Methods

For each experiment, three models (i.e. three different target pdfs) are considered: a model with $S = 0$ (Zero-Planets), a model with $S = 1$ (One-Planet) and a model with $S = 2$ (Two-Planets). The goal is to estimate the marginal likelihood of these models and then correctly detect the number of planets, i.e., $S = 0$ for **(E1)**, $S = 1$ for **(E2)** and $S = 2$ for **(E3)**. The marginal likelihoods corresponding to the Zero-planets models are available in closed form and need not be estimated (the model is simply Gaussian in that case). For this purpose, we apply NN-AQ (with $M = 10^7$) and and an IS procedure. We allocate a budget of $4 \cdot 10^6$ evaluations of the target. In IS, this budget is used to draw $4 \cdot 10^6$



(a) Zero planets (b) One planet (c) Two planets

Figure 4.6: Plot of marginal likelihood estimates of Model 1 (one-planet) and Model 2 (two-planets) versus σ for the three data sets. The straight lines represent the known marginal likelihoods of Model 0 (zero planets) for each data set. **(a)** data set with zero planets, **(b)** data set with one planet, **(c)** data set with two planets.

samples from the priors. While NN-AQ uses first $4 \cdot 10^6 - 5000$ of these samples to look for a good initialization, more specifically, the sample with the highest target evaluation is kept, along with 9 more samples taken at random, to use them as initial nodes. Then, NN-AQ is run for 5000 iterations. Both One-Planet and Two-Planets models are estimated for different values of $\sigma_e = 1, 2, \ldots, 15$. Note that we do not need to evaluate the target again when considering different σ_e, i.e., a single target evaluation can be reused for all values of σ_e. The results are shown in Figure 4.6.

Results

For each experiment **(E1)**-**(E3)**, Figure 4.6(a)-(c) depicts the estimations of Z of the different models provided by NN-AQ and IS, versus σ_e. The horizontal lines correspond to the known marginal likelihoods of Zero-Planets models. Overall, NN-AQ outperforms IS and predicts correctly the number of planets as well as the true value of σ_e (indeed, the curves corresponding to NN-AQ reach a maximum at $\sigma_e = 2$). Figure 4.6(a) shows that the estimations provided by NN-AQ and IS correctly rank the Zero-planets model ($S = 0$) as the most probable one. Figure 4.6(b) shows both NN-AQ and IS predict correctly the One-Planet model ($S = 1$) to be the correct one. However, for $\sigma_e = 2$, IS barely differentiates between the Zero-Planet and One-Planets models. Further, for $\sigma_e = 1$, it wrongly predicts Zero-Planets as the best one. Conversely, NN-AQ is able to predict the correct model for every value of σ_e, and besides, also predicts the true value $\sigma_e = 2$. In Figure 4.6(c), the difference in performance of NN-AQ and IS is more acute. While NN-AQ is able to correctly predict the Two-Planets model ($S = 2$) as the most probable for all values of σ_e, IS is unable to detect that second planet and, therefore, considers the One-Planet model more probable. As in the previous case, IS fails at detecting any planet for small values of σ_e. Again, NN-AQ predict the correct value of σ_e.

4.9. Conclusions

In this work, we have described a general framework for adaptive interpolative quadrature schemes, leveraging an in-depth study of different fields and related techniques in the literature, such as Bayesian quadrature algorithms, scattered data approximations, emulation, experimental design and active learning schemes. The nodes of the quadrature are adaptively chosen by maximizing a suitable acquisition function, which depends on the current interpolant and the positions of the nodes. This maximization does not require extra evaluations of the true posterior. The proposed methods supply also a surrogate model (emulator) which approximates the true posterior density, that can be also employed in further statistical analyses. Two specific schemes, based on Gaussian and NN bases, have been described. In both cases, a non-negative estimation $\widehat{Z}$ of the marginal likelihood Z is ensured.

In the proposed framework, we also relax the assumptions regarding the kernel-basis functions with respect to other approaches in the literature, e.g., the bases could be non-symmetric. For instance, the NN bases are non-symmetric functions and their use has different important benefits: **(a)** they ensure obtaining non-negative interpolation coefficients and estimators $\widehat{Z}$, **(b)** the linear system is directly solved without the need of inverting any matrix (the interpolation matrix is always diagonal), and **(c)** the bandwidth of the bases are automatically selected. Our scheme also allows selecting different kernel functions for each node point. Therefore, the quadrature rules in Bayesian quadrature are a special case of our proposed scheme. Indeed, Bayesian quadrature considers a single symmetric and semi positive definite kernel function. An importance sampling interpretation has been also provided. It is important to remark that the true posterior is only evaluated at the nodes selected sequentially by the algorithm, and the rest of other computations does not query the true model. The convergence of the proposed quadrature rules has been discussed, jointly with other theoretical results. The new algorithms are powerful techniques as also shown by several numerical experiments.

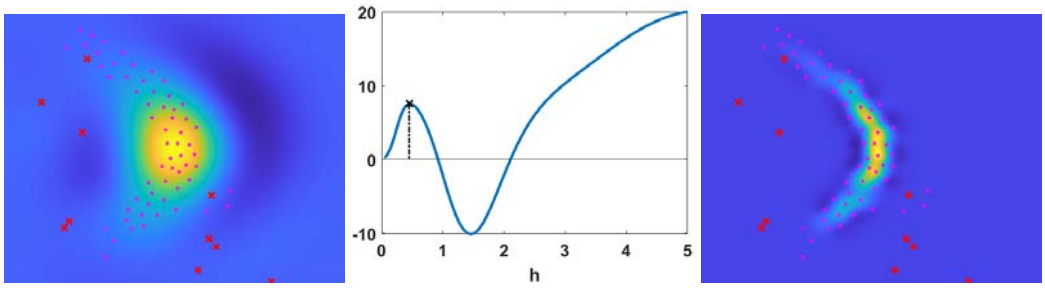
4.10. Appendix

4.10.1. Procedure for tuning the Gaussian kernel bandwidth

In this Appendix, we propose a procedure for fitting the bandwidth parameter h of the Gaussian kernel (GK),

$$k_G(\mathbf{x}, \mathbf{x}_i) = \frac{1}{(2\pi)^{\frac{d_x}{2}} h^{d_x}} \exp\left(-\frac{1}{2h^2}(\mathbf{x} - \mathbf{x}_i)^\top (\mathbf{x} - \mathbf{x}_i)\right), \tag{4.42}$$

when building the GK based interpolant of Sect. 4.3 for a given number of nodes. Assume we have run the GK-AQ algorithm (with some fixed h_0), so we have a total of N_T nodes. Now, for any h, we may solve the linear system (Eq. (4.5)), obtain the coefficients $\{\beta_i\}_{i=1}^{N_T}$



(a) (b) (c)

Figure 4.7: **(a)** GK based interpolant with $N_T = 70$ nodes and $h = 2.7$, fitted by maximizing the marginal likelihood. **(b)** Plot of $\widehat{Z}$ as function of h. The value of h at which $\widehat{Z}$ attains the local maximum is used to build the interpolant in our procedure. **(d)** GK based interpolant with $N_T = 70$ nodes and $h = 0.51$ fitted with the heuristic.

and calculate

$$\widehat{Z} = \sum_{i=1}^{N_T} \beta_i. \tag{4.43}$$

Note that, although not explicit, the β_i's, and hence $\widehat{Z}$, depend on h. The proposed procedure consists of taking h as the value where $\widehat{Z}$ attains its first local maximum. Starting from a small value h close to zero and increasing it, the estimation $\widehat{Z}$ is growing reaching a maximum. Then, h is starting to become "too big", producing too much overlapping among the kernel areas. The values of the elements out the diagonal of $\mathbf{K}$ grow, and some of the coefficients β_i are negative, and the estimation $\widehat{Z}$ decreases. As h becomes greater and greater, the matrix $\mathbf{K}$ tends to become ill-conditioned, and the absolute values of β_i's grows. Figure 4.7 compares the GK based interpolant of the target from Sect 4.8.1 with two different choices of h and $N_T = 70$ nodes. Figure 4.7(a) plots the interpolant taking h as the value which minimizes the marginal likelihood (see Sect. 4.3.2). Note that this value of h is too big given the dispersion of the nodes. While Figure 4.7(c) plots the interpolant taking h as the value where the curve of $\widehat{Z}$ (Figure 4.7(b)) attains its local maximum. This choice of h seems to fit better the existing nodes. Note also that, for some values of h, $\widehat{Z}$ may be negative.

4.10.2. Probabilistic interpretation of J

Let us consider $J = \int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$, which is the numerator of (4.1), our integral of interest I. In section 4.3.1, we have seen that, when $k(\mathbf{x}, \mathbf{x}_i) = k(\mathbf{x}_i, \mathbf{x})$ (i.e., a symmetric basis function), the interpolant $\widehat{\pi}(\mathbf{x}) = \sum_{i=1}^{N} \beta_i k(\mathbf{x}, \mathbf{x}_i)$ has the probabilistic interpretation of being the mean of the posterior distribution of (the "unknown") $\pi(\mathbf{x})$ after observing $\mathbf{d} = [\pi(\mathbf{x}_1), \ldots, \pi(\mathbf{x}_N)]^\top$, i.e., $\mathbb{E}[\pi(\mathbf{x})|\mathbf{d}] = \widehat{\pi}(\mathbf{x})$. The distribution on $\pi(\mathbf{x})$ induces a posterior distribution on J, which is a Gaussian with mean

$$\mathbb{E}[J|\mathbf{d}] = \widetilde{J} = \int_{\mathcal{X}} f(\mathbf{x})\widehat{\pi}(\mathbf{x})d\mathbf{x}, \tag{4.44}$$

and variance given by

$$\text{var}[J|\mathbf{d}] = \int\int k(\mathbf{x},\mathbf{x}')f(\mathbf{x})f(\mathbf{x}')d\mathbf{x}d\mathbf{x}' - \boldsymbol{\zeta}^\top \mathbf{K}^{-1}\boldsymbol{\zeta}, \tag{4.45}$$

where $\boldsymbol{\zeta} = [J_1, \ldots, J_N]$ and $J_i = \int_{\mathcal{X}} f(\mathbf{x})k(\mathbf{x},\mathbf{x}_i)d\mathbf{x}$. This interpretation corresponds to the so-called Bayesian quadrature, which uses Eq. (4.44) as approximation of J. Note that Eq. (4.44) is the quadrature obtained by substituting the true $\pi(\mathbf{x})$ with its interpolant $\widehat{\pi}(\mathbf{x})$, which coincides with the numerator of $\widehat{I}$ in Eq. (4.7).

4.10.3. Recursive inversion of a bordered matrix

The most costly step when calculating $\boldsymbol{\beta}$ in (4.5) consists in inverting the $N \times N$ matrix $(\mathbf{K})_{i,j} = k(\mathbf{x}_i,\mathbf{x}_j)$ $(i,j \in \{1,\ldots,N\})$. Moreover, every time a new node is added, the β_i must be recomputed, so the step of computing the inverse has to be done again. This time the matrix is bigger due to adding a new node, that is, it has an additional row and column. We show that knowing $\mathbf{K}^{-1}$ help us to compute the inverse of augmented matrices (called "bordered matrix", i.e., adding a "border" of new row and column to an existing matrix). Let us denote with $\mathbf{K}_N$ the matrix built using N nodes, and let $\mathbf{K}_{N+1}$ be the matrix with $N+1$ nodes. Of course we have

$$\mathbf{K}_{N+1} = \begin{pmatrix} \mathbf{K}_N & \mathbf{k}_N \\ \mathbf{k}_N^T & k \end{pmatrix} \tag{4.46}$$

where $\mathbf{k}_N = (k(\mathbf{x}_1,\mathbf{x}_{N+1}), k(\mathbf{x}_2,\mathbf{x}_{N+1}), \ldots, k(\mathbf{x}_N,\mathbf{x}_{N+1}))^T$ and $k = k(\mathbf{x}_{N+1},\mathbf{x}_{N+1})$. The $(N+1)\times(N+1)$ inverse of $\mathbf{K}_{N+1}$ can be expressed in terms of $\mathbf{K}_N^{-1}$ as follows

$$\mathbf{K}_{N+1}^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{c} & s \end{pmatrix}, \tag{4.47}$$

where

$$\begin{aligned}
\mathbf{A} &= \mathbf{K}_N^{-1} + \mathbf{K}_N^{-1}\mathbf{k}_N\left(k - \mathbf{k}_N^T\mathbf{K}_N^{-1}\mathbf{k}_N\right)^{-1}\mathbf{k}_N^T\mathbf{K}_N^{-1} \in \mathbb{R}^{N\times N},\\
\mathbf{b} &= -\mathbf{K}_N^{-1}\mathbf{k}_N\left(k - \mathbf{k}_N^T\mathbf{K}_N^{-1}\mathbf{k}_N\right)^{-1} \in \mathbb{R}^{N\times 1},\\
\mathbf{c} &= -\left(k - \mathbf{k}_N^T\mathbf{K}_N^{-1}\mathbf{k}_N\right)^{-1}\mathbf{k}_N^T\mathbf{K}_N^{-1} \in \mathbb{R}^{1\times N},\\
s &= \left(k - \mathbf{k}_N^T\mathbf{K}_N^{-1}\mathbf{k}_N\right)^{-1} \in \mathbb{R}.
\end{aligned}$$

Note that computing $s = \left(k - \mathbf{k}_N^T\mathbf{K}_N^{-1}\mathbf{k}_N\right)^{-1}$ is not costly since it is an scalar value.

4.10.4. Proofs

Proof to theorem 1

We have that

$$\begin{aligned} |J - \widehat{J}| &= \left| \int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} - \int_{\mathcal{X}} f(\mathbf{x})\widehat{\pi}(\mathbf{x})d\mathbf{x} \right| \\ &= \left| \int_{\mathcal{X}} f(\mathbf{x}) \left(\pi(\mathbf{x}) - \widehat{\pi}(\mathbf{x})\right) d\mathbf{x} \right|. \end{aligned}$$

It is easy to see that, for any $g(\mathbf{x})$ we have $-|g(\mathbf{x})| \leq g(\mathbf{x}) \leq |g(\mathbf{x})|$ for all $\mathbf{x}$, and that $-\int |g(\mathbf{x})|d\mathbf{x} \leq \int g(\mathbf{x})d\mathbf{x} \leq \int |g(\mathbf{x})|d\mathbf{x}$, so we have $\left|\int g(\mathbf{x})d\mathbf{x}\right| \leq \int |g(\mathbf{x})|d\mathbf{x}$. Using this result we can state the first inequality

$$\begin{aligned} |J - \widehat{J}| &= \left| \int_{\mathcal{X}} f(\mathbf{x}) \left(\pi(\mathbf{x}) - \widehat{\pi}(\mathbf{x})\right) d\mathbf{x} \right| \\ &\leq \int_{\mathcal{X}} |f(\mathbf{x})| \left|\pi(\mathbf{x}) - \widehat{\pi}(\mathbf{x})\right| d\mathbf{x} \\ &= \left\| f(\pi - \widehat{\pi}) \right\|_1. \end{aligned}$$

The second inequality of the theorem follows from Holder's inequality

$$\left\| f(\pi - \widehat{\pi}) \right\|_1 \leq \|f\|_2 \left\| \pi - \widehat{\pi} \right\|_2.$$

Finally, the last inequality of the theorem is obtained after manipulating the $\|f\|_2$ and $\left\|\pi - \widehat{\pi}\right\|_2$,

$$\begin{aligned} \|f\|_2 \left\| \pi - \widehat{\pi} \right\|_2 &= \left(\int_{\mathcal{X}} |f(\mathbf{x})|^2 d\mathbf{x} \right)^{\frac{1}{2}} \left(\int_{\mathcal{X}} |\pi(\mathbf{x}) - \widehat{\pi}(\mathbf{x})|^2 d\mathbf{x} \right)^{\frac{1}{2}} \\ &\leq \left(|\mathcal{X}| \max |f(\mathbf{x})|^2 \right)^{\frac{1}{2}} \left(|\mathcal{X}| \max |\pi(\mathbf{x}) - \widehat{\pi}(\mathbf{x})|^2 \right)^{\frac{1}{2}} \\ &= |\mathcal{X}| \max |f(\mathbf{x})| \max |\pi(\mathbf{x}) - \widehat{\pi}(\mathbf{x})| \\ &= |\mathcal{X}| \, \|f\|_\infty \left\| \pi - \widehat{\pi} \right\|_\infty. \end{aligned}$$

Proof to theorem 2

We provide the main concepts and elements of the proof. For more details, see [70, 7]. Let $J = \int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$ and $\widetilde{J} = \sum_{i=1}^{N} \nu_i \pi(\mathbf{x}_i)$ be the integral of interest and the quadrature using points $\{\mathbf{x}_i\}_{i=1}^{N}$, respectively. Recall that we also denote $\boldsymbol{\nu} = [\nu_1, \ldots, \nu_N]^\top$.
Consider that π is a function belonging to the reproducing kernel Hilbert space of functions $\mathcal{H}$ originated from the symmetric and positive definite kernel function $k(\mathbf{x}, \mathbf{x}')$. Hence, J and $\widetilde{J}$ are functionals over that RKHS

$$\begin{aligned} J[\pi] &= \int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \\ \widetilde{J}[\pi] &= \sum_{i=1}^{N} \nu_i \pi(\mathbf{x}_i), \quad \pi \in \mathcal{H}. \end{aligned}$$

where we write explicitly $J[\cdot]$ is the functional that integrates w.r.t. $f(\mathbf{x})$, while $\widetilde{J}[\cdot]$ is the functional that integrates w.r.t. the weighted sum $\sum_{i=1}^{N} \nu_i \delta_{\mathbf{x}_i}$, where $\delta_{\mathbf{x}_i}$ denotes the point evaluation in $\mathbf{x}_i$. The integration error associated with $\widetilde{J}$ is characterized by the norm, in the dual space $\mathcal{H}^*$, of the error functional

$$\left\| J - \widetilde{J} \right\|_{\mathcal{H}^*} = \sup_{\|\pi\|_{\mathcal{H}} \leq 1} \left| \widetilde{J}[\pi] - J[\pi] \right|, \tag{4.48}$$

where $\|\cdot\|_{\mathcal{H}}$ and $\|\cdot\|_{\mathcal{H}^*}$ denote the norm in $\mathcal{H}$ and $\mathcal{H}^*$ respectively. Eq. (4.48) is also called worst-case error (WCE). Define the functions

$$k_f(\mathbf{x}) = \int_{\mathcal{X}} f(\mathbf{x}') k(\mathbf{x}, \mathbf{x}') d\mathbf{x}', \tag{4.49}$$

and

$$k_{\widetilde{f}}(\mathbf{x}) = \sum_{i=1}^{N} \nu_i k(\mathbf{x}, \mathbf{x}_i), \tag{4.50}$$

where $k_f, k_{\widetilde{f}} \in \mathcal{H}$. These functions exist as consequence of $\int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}) f(\mathbf{x}) d\mathbf{x} < \infty$. It can be shown that $\left\| J - \widetilde{J} \right\|_{\mathcal{H}^*} = \left\| k_f - k_{\widetilde{f}} \right\|_{\mathcal{H}}$, and

$$\left\| J - \widetilde{J} \right\|_{\mathcal{H}^*}^2 = \boldsymbol{\nu}^\top \mathbf{K} \boldsymbol{\nu} - 2\boldsymbol{\nu}^\top \boldsymbol{\zeta} + \int_{\mathcal{X}} \int_{\mathcal{X}} f(\mathbf{x}) f(\mathbf{x}') k(\mathbf{x}, \mathbf{x}') d\mathbf{x} d\mathbf{x}', \tag{4.51}$$

for a vector of weights $\boldsymbol{\nu} \in \mathbb{R}^N$, the matrix $(\mathbf{K})_{1 \leq i,j \leq N} = k(\mathbf{x}_i, \mathbf{x}_j)$, and the vector of integrals $\boldsymbol{\zeta} = [k_f(\mathbf{x}_1), \ldots, k_f(\mathbf{x}_N)]^\top$. Conditional on the fixed states $\{\mathbf{x}_i\}_{i=1}^N$, the weights $\boldsymbol{\nu}$ that minimizes the above expression are given by $\boldsymbol{\nu} = \mathbf{K}^{-1}\boldsymbol{\zeta}$. These are the weights that arises if we build the interpolant $\widehat{\pi}$ of π at points $\{\mathbf{x}_i\}_{i=1}^N$, using $k(\mathbf{x}, \mathbf{x}')$ as the basis function, and substitute it in J to obtain the quadrature.

Proof to theorem 6

Let J be the integral of interest, and let $\tilde{J} = \sum_{i=1}^{N} \beta_i J_i$ and $\widehat{J} = \sum_{i=1}^{N} \beta_i \widehat{J}_i$ be the approximations using, respectively, the exact J_i and the noisy estimation $\widehat{J}_i$. Recall that the coefficients β_i are written in matrix form as $\boldsymbol{\beta} = \mathbf{K}^{-1}\mathbf{d}$ where $\mathbf{K}$ is the interpolation matrix and $\mathbf{d}$ is the vector of evaluations of π. Let us denote $\boldsymbol{\zeta} = [J_1, \ldots, J_N]^\top$ and $\widehat{\boldsymbol{\zeta}} = [\widehat{J}_1, \ldots, \widehat{J}_N]^\top$. Denoting the dot product in $\mathbb{R}^N$ as $\langle \cdot, \cdot \rangle$, we can express $\widetilde{J} = \langle \boldsymbol{\zeta}, \boldsymbol{\beta} \rangle$ and $\widehat{J} = \langle \widehat{\boldsymbol{\zeta}}, \boldsymbol{\beta} \rangle$. Thus

$$\begin{aligned}
|J - \widehat{J}| &= |J - \langle \widehat{\boldsymbol{\zeta}}, \boldsymbol{\beta} \rangle| \\
&= |J - \langle \boldsymbol{\zeta} - \boldsymbol{\zeta} + \widehat{\boldsymbol{\zeta}}, \boldsymbol{\beta} \rangle| \\
&= |J - \langle \boldsymbol{\zeta}, \boldsymbol{\beta} \rangle + \langle \boldsymbol{\zeta}, \boldsymbol{\beta} \rangle - \langle \widehat{\boldsymbol{\zeta}}, \boldsymbol{\beta} \rangle| \\
&\leq |J - \widetilde{J}| + |\langle \boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}, \boldsymbol{\beta} \rangle| \\
&= |J - \widetilde{J}| + |\langle \mathbf{K}^{-1}(\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}), \boldsymbol{d} \rangle| \\
&\leq \|f(\pi - \widehat{\pi})\|_1 + \|\mathbf{K}^{-1}(\boldsymbol{\zeta} - \widetilde{\boldsymbol{\zeta}})\|_2 \, \|\mathbf{d}\|_2 \\
&\leq |\mathcal{X}| \, \|f\|_\infty \, \|\pi - \widehat{\pi}\|_\infty + \|\mathbf{K}^{-1}\|_2 \|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\|_2 \, \|\mathbf{d}\|_2
\end{aligned}$$

where the norm $\|\mathbf{K}^{-1}\|_2$ represents the largest singular value of $\mathbf{K}^{-1}$. The bounds $\left\|\pi - \widehat{\pi}\right\|_\infty = \lambda(r)$ and $\|\mathbf{K}^{-1}\|_2 = O(\upsilon(s,h))$ for different RBF can be found respectively in Chapters 11.3 and 12.2 of [76]. For further details, see Proposition 1 in [70].

Proof to theorem 8

Let us consider the target $\pi(\mathbf{x})$ and the interpolant $\widehat{\pi}(\mathbf{x})$ based on NN constant kernels. Note that for all $\mathbf{x} \in \mathcal{X}$ we have $\widehat{\pi}(\mathbf{x}) = \pi(\mathbf{x}^*)$, where $\mathbf{x}^* = \arg\min_i \|\mathbf{x} - \mathbf{x}_i\|$, i.e., the node that is closest to $\mathbf{x}$. Lipschitz continuity implies that $|\pi(\mathbf{z}) - \pi(\mathbf{x})| \leq L_0 \|\mathbf{z} - \mathbf{x}\|$ for all $\mathbf{z}, \mathbf{x} \in \mathcal{X}$. Hence,

$$\begin{aligned}
\left\|\pi - \widehat{\pi}\right\|_\infty &= \max_{\mathbf{x}\in\mathcal{X}} |\pi(\mathbf{x}) - \widehat{\pi}(\mathbf{x})| \\
&= \max_{\mathbf{x}\in\mathcal{X}} |\pi(\mathbf{x}) - \pi(\mathbf{x}^*)| \\
&\leq L_0 \max_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x} - \mathbf{x}^*\| \\
&= L_0 \max_{\mathbf{x}\in\mathcal{X}} \min_i \|\mathbf{x} - \mathbf{x}_i\| \\
&= L_0 r,
\end{aligned}$$

where we used the definition of fill distance r, i.e.,

$$r = \max_{\mathbf{x}\in\mathcal{X}} \min_i \|\mathbf{x} - \mathbf{x}_i\| .$$

For further details, see [11, 60].

Bibliography

[1] L. Affer et al. HADES RV program with HARPS-N at the TNG. IX. A super-Earth around the M dwarf Gl 686. *arXiv:1901.05338*, 622:A193, February 2019.

[2] O .D. Akyildiz and J. Míguez. Nudging the particle filter. *Statistics and Computing*, 30:305–330, 2020.

[3] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[4] Y. Auffray, P. Barbillon, and J.-M. Marin. Maximin design on non hypercube domains and kernel interpolation. *Statistics and Computing*, 22(3):703–712, 2012.

[5] S. C. C Barros et al. WASP-113b and WASP-114b, two inflated hot Jupiters with contrasting densities. *Astronomy and Aastrophysics*, 593:A113, 2016.

[6] N. Bliznyuk, D. Ruppert, C. Shoemaker, R. Regis, S. Wild, and P. Mugunthan. Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. *Journal of Computational and Graphical Statistics*, 17(2):270–294, 2008.

[7] F.-X. Briol, C. J. Oates, M. Girolami, M. A. Osborne, and D. Sejdinovic. Probabilistic integration: A role in statistical computation? *Statistical Science*, 34(1):1–22, 2019.

[8] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric. Adaptive importance sampling: the past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017.

[9] R. L. Burden and J. D. Faires. *Numerical Analysis*. Brooks Cole, 2000.

[10] Daniel Busby. Hierarchical adaptive experimental design for Gaussian process emulators. *Reliability Engineering & System Safety*, 94(7):1183–1193, 2009.

[11] T. Butler, L. Graham, S. Mattis, and S. Walsh. A measure-theoretic interpretation of sample based numerical integration with applications to inverse and prediction problems under uncertainty. *SIAM Journal on Scientific Computing*, 39(5):A2072–A2098, 2017.

[12] R. E. Caflisch. Monte Carlo and Quasi–Monte Carlo methods. *Acta numerica*, 7:1–49, 1998.

[13] D. Chauveau and P. Vandekerkhove. Improving convergence of the Hastings–Metropolis algorithm with an adaptive proposal. *Scandinavian Journal of Statistics*, 29(1):13–29, 2002.

[14] Y. Chen, M. Welling, and A. Smola. Super-samples from kernel herding. *In Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 1–8, 2010.

[15] P. R. Conrad, Y. M. Marzouk, N. S. Pillai, and A. Smith. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516):1591–1607, 2016.

[16] B. Delyon, F. Portier, et al. Integral approximation by kernel smoothing. *Bernoulli*, 22(4):2177–2208, 2016.

[17] L. Devroye, L. Györfi, G. Lugosi, and H. Walk. On the measure of Voronoi cells. *Journal of Applied Probability*, 54(2):394–408, 2017.

[18] V. Elvira, L. Martino, and P. Closas. Importance Gaussian quadrature. *arXiv:2001.03090*, pages 1–13, 2020.

[19] V. Elvira, L. Martino, and P. Closas. Importance Gaussian Quadrature. *arXiv:2001.03090*, 2020.

[20] J. Felip, N. Ahuja, and O. Tickoo. Tree pyramidal adaptive importance sampling. *arXiv preprint arXiv:1912.08434*, 2019.

[21] W. R. Gilks, N. G. Best, and K. K. C. Tan. Adaptive Rejection Metropolis Sampling within Gibbs Sampling. *Applied Statistics*, 44(4):455–472, 1995.

[22] W. R. Gilks and P. Wild. Adaptive Rejection Sampling for Gibbs Sampling. *Applied Statistics*, 41(2):337–348, 1992.

[23] M. Golomb and H. F. Weinberger. Optimal approximations and error bounds. Technical report, Wisconsin Univ-Madison Mathematics Research Center, 1958.

[24] D. Görür and Y. W. Teh. Concave convex adaptive rejection sampling. *Journal of Computational and Graphical Statistics*, 20(3):670–691, 2011.

[25] Philip C. Gregory. Bayesian re-analysis of the Gliese 581 exoplanet system. *Monthly Notices of the Royal Astronomical Society*, 415(3):2523–2545, August 2011.

[26] T. Gunter, M. A. Osborne, R. Garnett, P. Hennig, and S. J. Roberts. Sampling for inference in probabilistic models with fast Bayesian quadrature. In *Advances in neural information processing systems*, pages 2789–2797, 2014.

[27] T. E. Hanson, J. V. D. Monteiro, and A. Jara. The Polya tree sampler: Toward efficient and automatic independent Metropolis–Hastings proposals. *Journal of Computational and Graphical Statistics*, 20(1):41–62, 2011.

[28] W. Hörmann. A rejection technique for sampling from T-concave distributions. *ACM Transactions on Mathematical Software*, 21(2):182–193, 1995.

[29] P. Jäckel. A note on multivariate Gauss–Hermite quadrature. *London: ABN-Amro. Re*, 2005.

[30] M. Järvenpää, M. U. Gutmann, A. Vehtari, P. Marttinen, et al. Parallel gaussian process surrogate bayesian inference with noisy likelihood evaluations. *Bayesian Analysis*, 2020.

[31] M. Kanagawa and P. Hennig. Convergence Guarantees for Adaptive Bayesian Quadrature Methods. In *Advances in Neural Information Processing Systems*, pages 6234–6245, 2019.

[32] K. Kandasamy, J. Schneider, and B. Póczos. Query efficient posterior estimation in scientific experiments via Bayesian active learning. *Artificial Intelligence*, 243:45–56, 2017.

[33] T. Karvonen, M. Kanagawa, and S. Särkkä. On the positivity and magnitudes of Bayesian quadrature weights. *Statistics and Computing*, 29(6):1317–1333, 2019.

[34] T. Karvonen and S. Särkkä. Classical quadrature rules via Gaussian processes. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2017.

[35] T. Karvonen and S. Sarkka. Fully symmetric kernel quadrature. *SIAM Journal on Scientific Computing*, 40(2):A697–A720, 2018.

[36] M. Kennedy. Bayesian quadrature with non-normal approximating functions. *Statistics and Computing*, 8(4):365–375, 1998.

[37] S. Lacoste-Julien, F. Lindsten, and F. Bach. Sequential kernel herding: Frank-Wolfe optimization for particle filtering. *In Proc. of the 18th International Conference on Artificial Intelligence and Statistics*, pages 544–552, 2015.

[38] Q. Liu and D. A. Pierce. A note on Gauss–Hermite quadrature. *Biometrika*, 81(3):624–629, 1994.

[39] Qiang Liu and Jason D. Lee. Black-box importance sampling. In *Artificial Intelligence and Statistics (AISTATS)*, 2017.

[40] F. Llorente, L. Martino, D. Delgado, and J. Lopez-Santiago. Marginal likelihood computation for model selection and hypothesis testing: an extensive review. *viXra:2001.0052*, 2019.

[41] David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

[42] S. Mak and V. R. Joseph. Support points. *(to appear) Annals of Statistics, arXiv:1609.01811*, pages 1–55, 2018.

[43] L. Martino, R. Casarin, F. Leisen, and D. Luengo. Adaptive independent sticky MCMC algorithms. *EURASIP Journal on Advances in Signal Processing*, 2018(1):5, 2018.

[44] L. Martino and V. Elvira. Compressed Monte Carlo for distributed Bayesian inference. *viXra: 1811.0505*, 2018.

[45] L. Martino, V. Elvira, D. Luengo, and J. Corander. Layered adaptive importance sampling. *Statistics and Computing*, 27(3):599–623, 2017.

[46] L. Martino, D. Luengo, and J. Míguez. Independent random sampling methods. *Springer*, 2018.

[47] L. Martino and J. Míguez. A generalization of the adaptive rejection sampling algorithm. *Statistics and Computing*, 21(4):633–647, July 2011.

[48] L. Martino, J. Read, and D. Luengo. Independent doubly adaptive rejection metropolis sampling within gibbs sampling. *IEEE Transactions on Signal Processing*, 63(12):3123–3138, 2015.

[49] L. Martino, H. Yang, D. Luengo, J. Kanniainen, and J. Corander. A fast universal self-tuned sampler within Gibbs sampling. *Digital Signal Processing*, 47:68 – 83, 2015.

[50] R. Meyer, B. Cai, and F. Perron. Adaptive rejection Metropolis sampling using Lagrange interpolation polynomials of degree 2. *Computational Statistics and Data Analysis*, 52(7):3408–3423, March 2008.

[51] W. G. Müller. Coffee-house designs. In *Optimum design 2000*, pages 241–248. Springer, 2001.

[52] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial Mathematics, 1992.

[53] H. Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63. Siam, 1992.

[54] A. O'Hagan. Bayes–Hermite quadrature. *Journal of statistical planning and inference*, 29(3):245–260, 1991.

[55] M. Osborne, R. Garnett, Z. Ghahramani, D. K. Duvenaud, S. J. Roberts, and C. E. Rasmussen. Active learning of model evidence using Bayesian quadrature. In *Advances in neural information processing systems*, pages 46–54, 2012.

[56] A. Owen. *Monte Carlo theory, methods and examples*. 2013.

[57] Matthew T Pratola, C Devon Lin, and Peter F Craigmile. Optimal design emulators: A point process approach. *arXiv preprint arXiv:1804.02089*, 2018.

[58] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C++ : the art of scientific computing*. Springer, 2002.

[59] L. Pronzato and W. G. Müller. Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22(3):681–701, 2012.

[60] Luc Pronzato. Minimax and maximin space-filling designs: some properties and methods for construction. *Journal de la Societe Française de Statistique*, 158(1):7–36, 2017.

[61] M. H. Protter, B. Charles Jr, et al. *A first course in real analysis*. Springer Science & Business Media, 2012.

[62] C. E. Rasmussen, J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A.F.M. Smith, and M. West. Gaussian processes to speed up hybrid Monte Carlo for expensive Bayesian integrals. In *Bayesian Statistics 7*, pages 651–659, 2003.

[63] C. E. Rasmussen and Z. Ghahramani. Bayesian Monte Carlo. *Advances in neural information processing systems*, pages 505–512, 2003.

[64] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[65] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[66] G. Santin and B. Haasdonk. Convergence rate of the data-independent P-greedy algorithm in kernel-based approximation. *Dolomites Research Notes on Approximation*, 10(Special_Issue), 2017.

[67] R. Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3(3):251–264, 1995.

[68] R. Schaback. *Reconstruction of multivariate functions from scattered data*. PhD thesis, Citeseer, 1997.

[69] R. Schaback. Native Hilbert spaces for radial basis functions i. In *New Developments in Approximation Theory*, pages 255–282. Springer, 1999.

[70] A. Sommariva and M. Vianello. Numerical cubature on scattered data by radial basis functions. *Computing*, 76(3-4):295, 2006.

[71] A. Stuart and A. Teckentrup. Posterior consistency for Gaussian process approximations of Bayesian posterior distributions. *Mathematics of Computation*, 87(310):721–753, 2018.

[72] D. H. Svendsen, L. Martino, and G. Camps-Valls. Active emulation of computer codes with gaussian processes - application to remote sensing. *Pattern Recognition*, 100:107103, 2020.

[73] Trifon Trifonov, Stephan Stock, Thomas Henning, Sabine Reffert, Martin Kürster, Man Hoi Lee, Bertram Bitsch, R. Paul Butler, and Steven S. Vogt. Two Jovian Planets around the Giant Star HD 202696: A Growing Population of Packed Massive Planetary Pairs around Massive Stars? *The Astronomical Journal*, 157(3):93, March 2019.

[74] L.M.M. van den Bos, B. Sanderse, and W.A.A.M. Bierbooms. Adaptive sampling-based quadrature rules for efficient bayesian prediction. *Journal of Computational Physics*, page 109537, 2020.

[75] H. Wang and J. Li. Adaptive Gaussian process approximation for Bayesian inference with expensive likelihood functions. *Neural computation*, 30(11):3072–3094, 2018.

[76] H. Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004.

[77] H. Ying, K. Mao, and K. Mosegaard. Moving Target Monte Carlo. *arXiv preprint arXiv:2003.04873*, 2020.

5. DEEP IMPORTANCE SAMPLING BASED ON REGRESSION FOR MODEL INVERSION AND EMULATION

In *Digital Signal Processing*, Volume 116, 103104 (2021)

F. Llorente[1], L. Martino[2], D. Delgado-Gómez[1], G. Camps-Valls[3]
[1] Dep. of Statistics, Universidad Carlos III de Madrid, Leganés (Spain).
[2] Dep. of Signal Processing, Universidad Rey Juan Carlos, Móstoles (Spain).
[3] Dep. of Electrical Engineering, Universitat de València, Valencia (Spain).

Abstract

Understanding systems by forward and inverse modeling is a recurrent topic of research in many domains of science and engineering. In this context, Monte Carlo methods have been widely used as powerful tools for numerical inference and optimization. They require the choice of a suitable proposal density that is crucial for their performance. For this reason, several adaptive importance sampling (AIS) schemes have been proposed in the literature. We here present an AIS framework called Regression-based Adaptive Deep Importance Sampling (RADIS). In RADIS, the key idea is the adaptive construction via regression of a non-parametric proposal density (i.e., *an emulator*), which mimics the posterior distribution and hence minimizes the mismatch between proposal and target densities. RADIS is based on a deep architecture of two (or more) nested IS schemes, in order to draw samples from the constructed emulator. The algorithm is highly efficient since employs the posterior approximation as proposal density, which can be improved adding more support points. As a consequence, RADIS asymptotically converges to an exact sampler under mild conditions. Additionally, the emulator produced by RADIS can be in turn used as a cheap surrogate model for further studies. We introduce two specific RADIS implementations that use Gaussian Processes (GPs) and Nearest Neighbors (NN) for constructing the emulator. Several numerical experiments and comparisons show the benefits of the proposed schemes. A real-world application in remote sensing model inversion and emulation confirms the validity of the approach.

Keywords: Model Inversion; Bayesian Inference; Emulation; Adaptive Regression; Importance Sampling; Sequential Inversion; Remote Sensing.

5.1. Introduction

Modeling and understanding systems is of paramount relevance in many domains of science and engineering. The problems involve both forward and inverse modeling, and very often one resorts to domain knowledge (either in the form of mechanistic models, hypotheses, constraints or just data) and observational data to learn parametrizations and

do inferences. Among the many approaches possible, Bayesian methods have become very popular during the last decades. Bayesian inference is very active in the communities of machine learning, statistics and signal processing [59, 49, 71]. With them, there has been a surge of interest in the Monte Carlo (MC) techniques that are often necessary for the implementation of the Bayesian analysis. Several families of MC schemes have been proposed that excel in numerous applications, including the popular Markov Chain Monte Carlo (MCMC) algorithms, particle filtering techniques and adaptive importance sampling (AIS) methods [71, 4].

Adaptive Importance Sampling (AIS). The performance of the MC algorithms depends strongly on the proper choice of a proposal probability density function (pdf). In adaptive schemes, the proposal pdf is updated considering the previous generated samples. In recent years, a plethora of AIS algorithms have been proposed in the literature [4]. In most of these algorithms, the complete proposal can be expressed as a finite parametric mixture of densities [10, 9, 20, 18, 47]. Unlike these schemes, we consider a non-parametric proposal based on an interpolating construction.

Emulators in Bayesian Inference. Furthermore, many Bayesian inference problems involve the evaluation of computationally intensive models, due to the use of particularly complex systems, consisting of many coupled ordinary or partial differential equations in high-dimensional spaces, or a large amount of available data. To overcome this issue, a successful approach consists in replacing the true model by a surrogate model (a.k.a. *an emulator*) [61, 5, 74, 70, 78].

The resulting emulator can be employed in different ways inside a Bayesian analysis. A first possibility is to apply MC sampling methods considering the surrogate model as an approximate posterior pdf within the MC schemes [11, 82, 13][38, Chapter 9.4.3] or within different quadrature rules [34, 67, 40], instead of the evaluation of a costly true posterior. For instance, this is also the case of the strategy known as *calibrate, emulate, sample*, currently in vogue [12]. In order to improve the efficiency of MC algorithms, a second option is to use the emulator as a proposal density within an MC technique. Here, we focus on the last approach.

Contribution. In this work, we design a deep AIS framework where a non-parametric interpolating proposal density is adapted online. The new approach is called Regression-based Adaptive Deep Importance Sampling (RADIS). In RADIS, the key idea is the adaptive construction of a non-parametric proposal pdf (i.e., *an emulator*), which mimics the posterior distribution in order to minimize the mismatch between proposal and target pdfs. Differently from other adaptive schemes, the adaptation in RADIS not only uses the information of the previous samples, but also all the evaluations of the posterior for directly constructing the emulator. Thus, unlike in a parametric approach, in our setting this discrepancy can be arbitrarily decreased to zero by adding more nodes. Hence, RADIS is asymptotically an exact sampler. The proposed methodology is based on a *deep architecture*: two nested IS schemes are employed, with an inner and an outer IS layers. The inner IS stage is used to generate samples from the emulator. The outer IS layer provides

the final posterior approximation by a cloud of weighted samples. Thus, RADIS finally provides two approximations of the posterior, one in form of a weighted particle measure, and also the emulator adapted online.[6] Parsimonious constructions of the emulator have been also discussed.

We discuss two specific implementation of RADIS. These specific implementations differ on the choice of the emulator construction. In the first one, a Gaussian Process (GP) model is applied to the log-posterior function obtaining the novel scheme denoted as GP-AIS. In the second one, a piece-wise constant approximation based on Nearest Neighbors (NNs) is applied, providing the novel algorithm denoted as NN-AIS. In both cases, the resulting proposal pdf can be seen as an *incremental mixture* of densities. A deep structure with more than two layers is described, where a chain of emulators is adapted and then employed as proposal pdfs within different nested IS stages. Robust and sequential implementations are also discussed. Several numerical comparisons show the advantages of RADIS with respect to benchmark algorithms. A real-word application illustrates the capabilities for sequential parameter retrieval and emulation of a well-known radiative transfer model (RTM) used in remote sensing. In the next section, a brief overview of the related approaches is provided.

5.2. Other related works

The non-parametric interpolating construction of the proposal and related strategies are appealing from different points of views. This is proved by attention devoted by the previous attempts in the literature shown above, and by other related approaches that we describe next.

Interpolating proposal. The idea of using interpolating densities is particularly attractive since we can arbitrarily decrease the mismatch between proposal and posterior by adding more support points. For this reason, the resulting algorithms provide very good performance [26, 55, 53, 54, 44]. The first use of an interpolating procedure for building a proposal density can be ascribed to the rejection sampling and adaptive rejection sampling schemes [27, 32, 29]. The well-know Zigurrat algorithm and table methods are other examples of fast rejection samplers employing interpolating proposals [42, 50]. They are state-of-the-art methods as random sample generators of specific univariate distributions in terms of speed of generation. In some rejection samplers and MCMC algorithms, the proposal is formed by polynomial pieces (constant, linear, etc.) [26, 55, 53, 54], [50, Chapters 4 and 7]. The use of interpolating proposal pdfs within an IS scheme is also considered in [22]. The conditions needed for applying an emulator as a proposal density are discussed in [44]. More specifically, we need to be able to: **(a)** update the construction of the emulator, **(b)** evaluate the emulator, **(c)** normalize the function defined by the emulator, and **(d)** draw samples from the emulator. It is not straightforward to find an interpolating (or regression) construction which satisfies all those conditions jointly, and

[6] The emulation can be applied to the entire posterior or part of it, like a physical model.

especially for an arbitrary dimensionality of the problem. This is the reason why the previous attempts of using an interpolating proposal pdfs are restricted to the univariate case. Our deep architecture solves these issues.

Partitioning and stratification. Note also that the use of a proposal pdf formed by components restricted to disjoint regions of the domain (like in the piecewise constant proposal based on NN) is related to the stratification idea. Indeed, different schemes based on partitioning and/or stratification divide the entire domain in disjoint sub-regions and consider different partial proposals in each of them [71, Chapter 4.6.3], [36, 24, 65, 41]. The complete proposal pdf is then a mixture of the partial proposals. Moreover, this process can be iterated so that the partition is refined over the iterations increasing the number of partial proposals. In this case, the complete proposal is an *incremental mixture* as RADIS (see also below) [36, 41]. Recent works propose using trees in order to partition the space and subsequently build the proposal [22, 23]. In the context of MCMC, [31] builds an approximation of the target using Polya trees.

Incremental mixtures. The use of non-parametric but *non-interpolating* proposals have been suggested in other works. A non-parametric IS approach is considered in [83], where the proposal is built by a kernel density estimation. In [76], a proposal pdf defined as a mixture with increasing number of components is also suggested. When a weighting strategy based on the so-called *temporal deterministic mixture* is applied [48, 21], incremental mixture proposals appear also in other IS schemes (e.g., [48, 14]).

Other approaches. Surrogate GP models has been also employed within IS schemes in the context of rare event estimation [2, 17]. Finally, other IS schemes can be encompassed in a similar "deep" approach [57, 19]. In the first one, MCMC steps are used to jump from different tempered versions of the posterior, and a global IS weighting as product of intermediate weights [57]. In the second scheme a two-stages weighting procedure is used, where the first layer considers a Gauss-Hermite quadrature and the second layer is a standard IS method [19].

5.3. Preliminaries and motivation

5.3.1. Problem statement

Bayesian inference. In many real world applications, the goal is to infer a variable of interest given a set of data [60]. Let us denote the parameter of interest (static or dynamic) by $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$, and let $\mathbf{y} \in \mathbb{R}^{d_y}$ be the observed data. In a Bayesian analysis, all the statistical information is contained in the posterior distribution, which is given by

$$\bar{\pi}(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{Z(\mathbf{y})}, \tag{5.1}$$

where $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf, and $Z(\mathbf{y})$ is the Bayesian model evidence (a.k.a. marginal likelihood). The marginal likelihood $Z(\mathbf{y})$ is important for model selection purposes [39, 52]. Generally, $Z(\mathbf{y})$ is unknown, so we are able to

evaluate the unnormalized target function, $\pi(\mathbf{x}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})$. The analytical computation of the posterior density $\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x})$ is often unfeasible, hence numerical approximations are needed. Our goal is to approximate integrals of the form

$$I = \int_{\mathcal{X}} f(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x} = \frac{1}{Z}\int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \tag{5.2}$$

where $f(\mathbf{x})$ is some integrable function, and

$$Z = \int_{\mathcal{X}} \pi(\mathbf{x})d\mathbf{x}. \tag{5.3}$$

In the literature, random sampling or deterministic quadratures are often used [71, 50, 58]. In this work, we focus on the so-called IS approach.

Emulation. There exist many situations where the evaluation of π is expensive (e.g., as in big data framework or when the observation model is costly). Hence, we are also interested in obtaining an emulator of $\pi(\mathbf{x})$ (or just a part of the posterior), denoted $\widehat{\pi}_t(\mathbf{x})$, such that (i) $\widehat{\pi}_t(\mathbf{x})$ is cheap to evaluate, and (ii) $\widehat{\pi}_t(\mathbf{x}) \to \pi(\mathbf{x})$ (in some sense, e.g., L_2 norm) as $t \to \infty$.

5.3.2. Importance sampling (IS) and aim of the work

Let us consider a normalized proposal density $\bar{q}(\mathbf{x})$.[7] The importance sampling (IS) method consists of drawing N independent samples, $\mathbf{x}_1, \ldots, \mathbf{x}_N$, from $\bar{q}(\mathbf{x})$ (also called particles), and then assign to each sample the following unnormalized weights

$$w_n = w(\mathbf{x}_n) = \frac{\pi(\mathbf{x}_n)}{\bar{q}(\mathbf{x}_n)}, \quad n = 1, \ldots, N. \tag{5.4}$$

An unbiased estimator of the marginal likelihood Z is given by the arithmetic mean of these unnormalized weights [37, 71], i.e.,

$$\widehat{Z} = \frac{1}{N}\sum_{n=1}^{N} w_n.$$

Defining also the normalized weights $\bar{w}_n = \frac{w_n}{\sum_{i=1}^{N} w_i}$, with $n = 1, \ldots, N$, the self-normalized IS estimator of I in Eq. (5.2) is given by

$$\widehat{I} = \sum_{n=1}^{N} \bar{w}_n f(\mathbf{x}_n).$$

More generally, regardless of the specific function $f(\mathbf{x})$, we obtain a particle approximation of $\bar{\pi}$, i.e., $\widetilde{\pi}(\mathbf{x}) = \sum_{n=1}^{N} \bar{w}_n \delta(\mathbf{x} - \mathbf{x}_n)$, where $\delta(\mathbf{x})$ is a delta function. It is important to remark that with this particle approximation, we can approximate several quantities related to the posterior $\bar{\pi}(\mathbf{x})$, such as any moments and/or credible intervals (not just a specific integral). The quality of this particle approximation is related to the discrepancy

[7] We assume that $\bar{q}(\mathbf{x}) > 0$ for all $\mathbf{x}$ where $\bar{\pi}(\mathbf{x}) > 0$, and $\bar{q}(\mathbf{x})$ has heavier tails than $\bar{\pi}(\mathbf{x})$.

between the proposal $\bar{q}(\mathbf{x})$ and the posterior $\bar{\pi}(\mathbf{x})$. Indeed, in an ideal MC scenario, we can draw from the posterior, i.e., $\bar{q}(\mathbf{x}) = \bar{\pi}(\mathbf{x})$, so that $\bar{w}_n = \frac{1}{N}$, which corresponds with the maximum effective sample size (ESS) [37, 46]. With a generic proposal $\bar{q}(\mathbf{x}) \propto \bar{q}(\mathbf{x})$, we can obtain a very small ESS and a bad particle approximation $\widetilde{\pi}(\mathbf{x})$ (i.e., poor performance of the algorithm).

Remark 1. *The variance of the marginal likelihood estimator* $\widehat{Z} = \frac{1}{N}\sum_{n=1}^{N} w(\boldsymbol{x}_n)$ *is given by*

$$\mathtt{var}[\widehat{Z}] = \frac{1}{N}\mathtt{var}[w(\boldsymbol{x})], \tag{5.5}$$

where $w(\boldsymbol{x}) = \frac{\pi(\mathbf{x})}{\bar{q}(\mathbf{x})}$ *and* $\boldsymbol{x} \sim \bar{q}(\boldsymbol{x})$. *Since* $\widehat{Z}$ *is also unbiased, then we also have*

$$\mathbb{E}[|Z - \widehat{Z}|^2] = \frac{1}{N}\mathtt{var}[w(\boldsymbol{x})]. \tag{5.6}$$

For more details, see [71].

Remark 2. *The variance of the IS weight function* $w(\boldsymbol{x})$ *is proportional to the Pearson divergence between* $\bar{q}(\boldsymbol{x})$ *and* $\bar{\pi}(\boldsymbol{x})$, *denoted as* $\chi^2(\bar{\pi}\|\bar{q})$ *(also called* χ^2 *distance), i.e.,*

$$\mathtt{var}[w(\boldsymbol{x})] \propto \chi^2(\bar{\pi}\|\bar{q}) = \int_{\mathcal{X}} \frac{(\bar{\pi}(\boldsymbol{x}) - \bar{q}(\boldsymbol{x}))^2}{\bar{q}(\boldsymbol{x})} d\boldsymbol{x}. \tag{5.7}$$

See [46, 1] and 5.10.1 for further details. Regarding the mean squared error of the estimator $\widehat{I}$, *we have*

$$\mathbb{E}[|I - \widehat{I}|^2] \leq \frac{C_f}{N}(\chi^2(\bar{\pi}\|\bar{q}) + 1). \tag{5.8}$$

The relationships with the L_2 *and* L_∞ *distances is also given in 5.10.1.*

To reduce the discrepancy between the proposal $\bar{q}(\mathbf{x})$ and the posterior $\bar{\pi}(\mathbf{x})$, we consider a non-parametric adaptive construction of the proposal $\bar{q}_t(\mathbf{x})$ where t denotes a discrete iteration index. In order to make that the discrepancy becomes smaller and smaller, an interpolating procedure $\bar{q}_t(\mathbf{x})$ based on a set of support points $\mathcal{S}_{t-1}$ is employed. Namely, we generate a sequence of proposal pdfs $\bar{q}_t(\mathbf{x})$, $\bar{q}_{t+1}(\mathbf{x})$, $\bar{q}_{t+2}(\mathbf{x})$,... which become closer and closer to $\bar{\pi}(\mathbf{x})$, as the number of support points grows. Throughout the paper, we denote $\bar{q}_t(\mathbf{x}) \propto \widehat{\pi}_t(\mathbf{x})$ the non-parametric regression function which approximates the unnormalized posterior $\pi(\mathbf{x})$ at iteration t. The normalized proposal is denoted as $\bar{q}_t(\mathbf{x}) = \frac{1}{c_t}\widehat{\pi}_t(\mathbf{x})$, where $c_t = \int_{\mathcal{X}} \widehat{\pi}_t(\mathbf{x}) d\mathbf{x}$. Although the approximation $\widehat{\pi}_t$ depends on the set of nodes $\mathcal{S}_{t-1}$, for simplicity we use the simpler notation $\widehat{\pi}_t(\mathbf{x}) = \widehat{\pi}_t(\mathbf{x}; \mathcal{S}_{t-1})$.

Remark 3. *If the sequence of proposals is such* $\left\|\bar{\pi} - \bar{q}_t\right\|_2 \to 0$ *as* $t \to \infty$, *then* $\chi^2(\bar{\pi}\|\bar{q}_t) \to 0$. *See 5.10.1 for more details.*

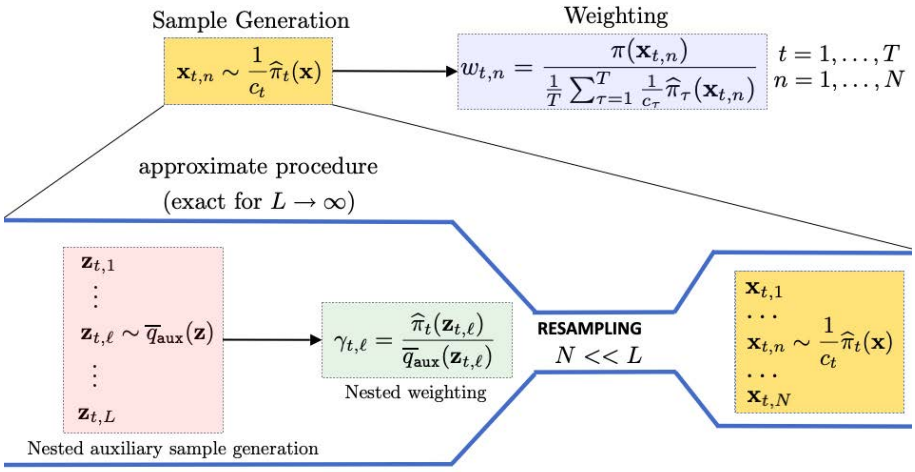


Figure 5.1: Approximate sampling from $\frac{1}{c_t}\widehat{\pi}_t(\mathbf{x}) \propto \widehat{\pi}_t(\mathbf{x})$ and final weighting scheme.

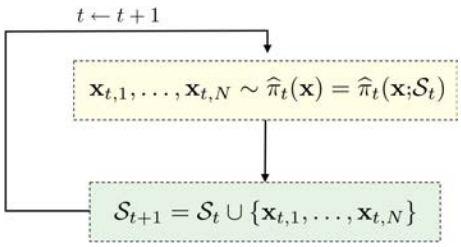


Figure 5.2: Graphical representation of the adaptation scheme. More parsimonious alternatives are introduced in Section 5.6.

5.4. Regression-based Adaptive Deep Importance Sampling

In this section, we introduce the proposed scheme, called Regression-based Adaptive Deep Importance Sampling (RADIS). The resulting algorithm is an adaptive importance sampler with a non-parametric interpolating proposal pdf. We show how to implement the sampling and construction of the proposal density in Sect. 5.4.1 and Sect. 5.4.2 respectively. The novel scheme is summarized in Table 5.1. The proposal is adaptively built using a regression approach that considers the set of all previous nodes $\mathbf{x}_i$'s where π is evaluated. In Section 5.4.2, we present two construction methodologies considered in this work. Samples from this proposal are drawn via an approximate procedure that can be interpreted as an additional "inner" IS. All the samples generated in the inner IS are then used in the "outer" IS. Figure 5.1 outlines this procedure. The adaptation consists in sequentially adding the samples to the set of current nodes (see Figure 5.2). In the outer IS, we consider a temporal deterministic mixture approach to compute the weights. Note that the weighting step needs to be done only once at the end of the algorithm.

5.4.1. RADIS: a two-layer Deep IS

RADIS is an adaptive IS scheme based on two IS stages. In the following, we describe the inner and outer stages as well as the possible construction and adaptation of the non-parametric proposal density. The extension with more than two nested layers is also discussed.

Inner IS scheme

The inner IS stage is repeated at every iteration. It generates samples approximately distributed from the current non-parametric proposal, denoted as $\widehat{\pi}_t$ (the unnormalized version). Furthermore, these samples are used to normalize $\widehat{\pi}_t$, i.e., in order to estimate $c_t = \int_{\mathcal{X}} \widehat{\pi}_t(\mathbf{x}) d\mathbf{x}$.

Approximate sampling from the emulator. It is not straightforward to sample from an interpolating proposal [26, 44]. We propose using an approximate procedure based on IS. Specifically, at each iteration, in order to sample from $\frac{1}{c_t}\widehat{\pi}_t(\mathbf{x})$, we use sampling importance resampling (SIR) with an auxiliary proposal $\bar{q}_{\text{aux}}$ [73]. First, a set of $\{\mathbf{z}_{t,\ell}\}_{\ell=1}^{L}$ (with large L) are drawn from $\bar{q}_{\text{aux}}(\mathbf{x})$. These auxiliary samples are weighted according to $\widehat{\pi}_t(\mathbf{x})$

$$\gamma_{t,\ell} = \frac{\widehat{\pi}_t(\mathbf{z}_{t,\ell})}{\bar{q}_{\text{aux}}(\mathbf{z}_{t,\ell})} \qquad \ell = 1, \dots, L.$$

Finally, in other to obtain $\{\mathbf{x}_{t,n}\}_{n=1}^{N}$, we resample N times within $\{\mathbf{z}_{t,\ell}\}_{\ell=1}^{L}$ with probabilities $\{\bar{\gamma}_{t,\ell}\}_{\ell=1}^{L}$ where $\bar{\gamma}_{t,\ell} = \frac{\gamma_{t,\ell}}{\sum_{i=1}^{L}\gamma_{t,i}}$ for $\ell = 1, ..., L$, i.e.,

$$\mathbf{x}_{t,n} \sim \sum_{\ell=1}^{L} \bar{\gamma}_{t,\ell}\delta(\mathbf{x} - \mathbf{z}_{t,\ell}), \qquad \text{for all} \quad n. \tag{5.9}$$

In this way, we obtain a set of samples $\{\mathbf{x}_{t,n}\}_{n=1}^{N}$ approximately distributed from $\widehat{\pi}_t$ [73, 75].

Remark 4. *Under some mild conditions, as $L \to \infty$, the SIR procedure is asymptotically exact. Namely, as $L \to \infty$ the density of the resampled particles becomes closer and closer to $q_t(\boldsymbol{x}) \propto \widehat{\pi}_t(\boldsymbol{x})$. See, for instance, the following references [73], [28, Sect. 6.2.4], [75, Sect. 3.2]. For further details, see [72, page 6], [45, App. A] and also 5.10.1.*

Remark 5. *Note that the computation of the inner IS weights $\gamma_{t,\ell}$'s does not involve the evaluation of the posterior $\pi(\boldsymbol{x})$, but only the evaluation of the emulator $\widehat{\pi}_t(\boldsymbol{x})$. Hence, assuming that the evaluation of the posterior is the main computational bottleneck, in this setting we can make L arbitrarily large.*

Since we resample from a finite set, we can obtain duplicated samples, but it rarely happens when $L >> N$. An alternative to avoid these repetitions is to use a regularized resampling, i.e.,

$$\mathbf{x}_{t,n} \sim \sum_{\ell=1}^{L} \bar{\gamma}_{t,\ell}K(\mathbf{x} - \mathbf{z}_{t,\ell}), \qquad \text{for all} \quad n, \tag{5.10}$$

where the deltas have been replaced by a kernel function $K(\mathbf{x})$ [56]. The bandwidth of $K(\mathbf{x})$ can tuned according to some kernel density estimation (KDE) criterion. For the computation of the outer IS weights (see below), we need to approximate $c_t = \int_{\mathcal{X}} \widehat{\pi}_t(\mathbf{x})d\mathbf{x}$ for $t = 1, \dots, T$. They are estimated during the inner IS by the corresponding estimator, $\widehat{c}_t = \frac{1}{L}\sum_{\ell=1}^{L} \gamma_{t,\ell}$, for $t = 1, \dots, T$. We have $\widehat{c}_t \to c_t$ when $L \to \infty$, by standard IS arguments [71].

Adaptation

At each iteration, at the end of the inner IS stage, the algorithm performs the adaptation producing $\widehat{\pi}_{t+1}$. Specifically, the emulator $\widehat{\pi}_t(\mathbf{x})$ is improved by incorporating the generated samples at each iteration as additional nodes (see Fig. 5.2). Namely, the additional support points $\{\mathbf{x}_{t,n}\}_{n=1}^{N}$ to $\mathcal{S}_t$ are obtained by resampling N times within $\{\mathbf{z}_{t,\ell}\}_{\ell=1}^{L}$ according to the probabilities $\bar{\gamma}_{t,\ell} = \frac{\gamma_{t,\ell}}{\sum_{i=1}^{L}\gamma_{t,i}}$ for $\ell = 1, ..., L$. Note that the probability mass $\bar{\gamma}_{t,\ell}$ is directly proportional to $\widehat{\pi}_t(\mathbf{z}_{t,\ell})$. Therefore, the algorithm tends to add points where $\widehat{\pi}_t$ is higher. Indeed, as $L \to \infty$, the resampled particles are distributed as $\widehat{\pi}_t$ [73, 75, 28]. If L is not great enough, some $\mathbf{x}_{t,n}$ can be repeated. We do not include these repetitions as support points. Increasing L or using a regularized resampling as in Eq. (5.10) avoids this issue [56]. Note that the number of support points $J_t = |\mathcal{S}_t|$ increases as t grows.
All the evaluations of the unnormalized posterior $\pi(\mathbf{x})$ in the additional nodes are stored in the vector denoted as $\boldsymbol{\pi}_t$, in order to be used in the outer IS stage. Note also that all evaluations of π are used to build the emulator.

Outer IS scheme

At the end of the iterative part, we compute the final IS weights $w_{t,n}$, using all the posterior evaluations $\pi_{t,n} = \pi(\mathbf{x}_{t,n})$, which are stored in the inner layer. More specifically, we assign to each sample (drawn also in the inner stage) the weight

$$w_{t,n} = \frac{\pi_{t,n}}{\frac{1}{T}\sum_{\tau=1}^{T}\frac{1}{\widehat{c}_\tau}\widehat{\pi}_\tau(\mathbf{x}_{t,n})}, \qquad \text{for all} \quad t = 1, ..., T, \quad n = 1, ..., N, \tag{5.11}$$

where we have employed a deterministic mixture weighting scheme [81, 21], i.e., the denominator consists of a temporal mixture (e.g., as also suggested in [14]). Note that the weights $w_{t,n}$ are not required in the iterative inner layer described above. Hence, they can be computed after the adaptation and sampling steps are finalized. The output of the algorithm is then formed by all the sets of weighted particles $\{\mathbf{x}_{t,n}, w_{t,n}\}_{n=1}^{N}$ for $t = 1, ..., T$, and the final emulator $\widehat{\pi}_{T+1}(\mathbf{x}) = \widehat{\pi}_{T+1}(\mathbf{x}; \mathcal{S}_T)$.

Remark 6. *As $t \to \infty$ and $L \to \infty$, then $\widehat{c}_t \to c_t \to Z$, i.e., is an approximation of the marginal likelihood. Another estimator of the marginal likelihood Z provided by RADIS is the arithmetic mean of all the outer weights, i.e., $\widehat{Z} = \frac{1}{NT}\sum_{t=1}^{T}\sum_{n=1}^{N} w_{t,n}$.*

Remark 7. *Additional layers can be included in the proposed deep architecture would consists in adapting a chain of several emulators. This is graphically represented in*

Table 5.1: **Regression-based Adaptive Deep Importance Sampling (RADIS)**

- Initialization: Choose the initial set $\mathcal{S}_0$ of nodes, and the values T, L, N (with $L >> N$). Obtain the vector of initial evaluations $\boldsymbol{\pi}_0$.
- For $t = 1, \ldots, T$**:**

1. **Emulator construction:** Given the set $\mathcal{S}_{t-1}$ and the corresponding vector of posterior evaluations $\boldsymbol{\pi}_{t-1}$, build the proposal function $\widehat{\pi}_t(\mathbf{x}) = \widehat{\pi}_t(\mathbf{x}|\mathcal{S}_{t-1})$ with a non-parametric regression procedure (see Sect. 5.4.2).

2. **Inner IS:**

   (a) *IS.* Sample $\{\mathbf{z}_{t,\ell}\}_{\ell=1}^{L} \sim q_{\text{aux}}(\mathbf{x})$ and compute the following weights

   $$\gamma_{t,\ell} = \frac{\widehat{\pi}_t(\mathbf{z}_{t,\ell})}{q_{\text{aux}}(\mathbf{z}_{t,\ell})}, \tag{5.12}$$

   for $\ell = 1, \ldots, L$.

   (b) *Resampling.* Resample $\{\mathbf{x}_{t,n}\}_{n=1}^{N}$ from $\{\mathbf{z}_{t,\ell}\}_{\ell=1}^{L}$ with probabilities $\{\bar{\gamma}_{t,\ell}\}_{\ell=1}^{L}$ where $\bar{\gamma}_{t,\ell} = \frac{\gamma_{t,\ell}}{\sum_{i=1}^{L}\gamma_{t,i}}$ for $\ell = 1, ..., L$.

   (c) *Normalizing constant.* Compute

   $$\widehat{c_t} = \frac{1}{L}\sum_{\ell=1}^{L}\gamma_{t,\ell}. \tag{5.13}$$

3. **Update:** Evaluate $\pi_{t,n} = \pi(\mathbf{x}_{t,n})$, for all $n = 1, ..., N$, and update the set of nodes appending $\mathcal{S}_t = \mathcal{S}_{t-1} \cup \{\mathbf{x}_{t,1}, ..., \mathbf{x}_{t,N}\}$ and $\boldsymbol{\pi}_t = [\boldsymbol{\pi}_{t-1}, \pi_{t,1}, ..., \pi_{t,N}]^\top$.

- Outer IS: Assign to each sample the weight

$$w_{t,n} = \frac{\pi_{t,n}}{\frac{1}{T}\sum_{\tau=1}^{T}\frac{1}{\widehat{c}_\tau}\widehat{\pi}_\tau(\mathbf{x}_{t,n})}, \qquad \text{for all} \quad t = 1, ..., T, \quad n = 1, ..., N.$$

- Outputs: Final emulator $\widehat{\pi}_{T+1}(\mathbf{x}) = \widehat{\pi}_{T+1}(\mathbf{x}|\mathcal{S}_T)$, and the set of weighted particles $\{\mathbf{x}_{t,n}, w_{t,n}\}_{n=1}^{N}$ for $t = 1, ..., T$.

Figure 5.3. One of the advantages of this deep approach with $D + 1 > 2$ layers (where D is the number of inner nested stages), is that different emulator constructions can be jointly applied. Each emulator serves as proposal of the next IS stage. In the additional layers, the evaluation of the posterior (true model) is not required. In this scenario, RADIS also provides D different emulators.

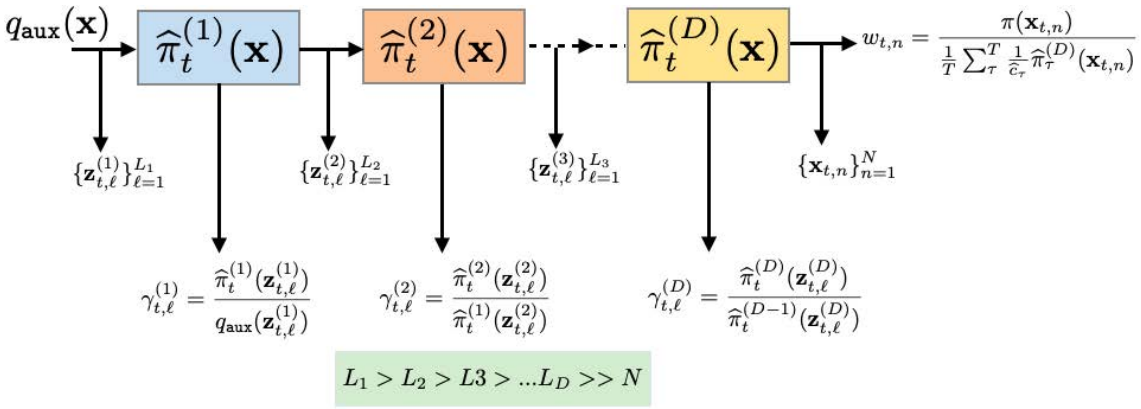


Figure 5.3: RADIS with $D+1$ layers in the deep architecture. Different emulator construction can be applied at each stage. In each d-th layer, the resampling is applied L_{d+1} times for generating the next cloud of resampled particles $\{\mathbf{z}_{t,\ell}^{(d+1)}\}_{\ell=1}^{L_{d+1}}$ (with $d = 1, ..., D+1$). These samples are used for the adaptation of $\widehat{\pi}_t^{(d)}$ and then are weighted again in the next stage. Note that $L_{D+1} = N$ and $L_d > L_{d+1}$.

5.4.2. Construction of $\widehat{\pi}$ by regression

We consider two different procedures to build the non-parametric proposal: a Gaussian process (GP) model and nearest neighbors (NN) scheme. In 5.10.1, we show that these constructions converges to the true underlying function as the number of nodes ($J_t = |\mathcal{S}_t|$) grows.

GP construction. Let us consider building the surrogate $\widehat{\pi}$ with Gaussian process (GP) regression in the log domain, i.e., over the $\log \pi(\mathbf{x})$ [68, 26]. GP regression provides with an approximation of a function from a set $\mathbf{x}_1, \ldots, \mathbf{x}_{J_t} \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$ (where $\mathcal{X}$ can be unbounded) and their corresponding function evaluation [69, 51]. To ensure the non-negativity of the approximation, we fit the GP to $\log \pi$ rather than directly on π [62]. Let $\phi(\mathbf{x}) \equiv \log \pi(\mathbf{x})$ and $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_{J_t}]^\top$ where $\phi_i = \log \pi(\mathbf{x}_i)$ for $i = 1, \ldots, J_t$. Given a symmetric and positive definite kernel $k(\mathbf{x}, \mathbf{x}')$ and some noise level σ, under the assumption that $\phi(\mathbf{x})$ is a zero-mean GP with kernel k, the GP regression of $\phi(\mathbf{x})$ is of the form

$$\widehat{\phi}_t(\mathbf{x}) = \sum_{i=1}^{J_t} \beta_i k(\mathbf{x}, \mathbf{x}_i), \tag{5.14}$$

where the coefficients $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{J_t}]^\top$ are given by

$$\boldsymbol{\beta} = (\mathbf{K} + \zeta \mathbf{I})^{-1} \boldsymbol{\phi} \tag{5.15}$$

with $(\mathbf{K})_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ for $1 \le i, j \le J_t$ and $\mathbf{I}$ is the $J_t \times J_t$ identity matrix. Note that, for $\zeta = 0$, $\widehat{\phi}$ corresponds to an interpolator of ϕ. Note also that the cost of obtaining $\widehat{\phi}$ is $O({J_t}^3)$ since it requires inverting a $J_t \times J_t$ matrix. As an example, a possible choice of kernel is the Gaussian $k(\mathbf{x}, \mathbf{x}') = \exp\{-\frac{1}{2\epsilon^2} \|\mathbf{x} - \mathbf{x}'\|_2^2\}$, where the hyperparameter ϵ can be estimated, e.g., by maximizing the marginal likelihood [68]. Finally, the approximation

of π is given by

$$\widehat{\pi}_t(\mathbf{x}) = \exp\{\widehat{\phi}_t(\mathbf{x})\}. \tag{5.16}$$

Instead of building on the emulator in the log-domain, a simpler alternative (to ensure non-negativity) consists in setting $\phi(\mathbf{x}) \equiv \pi(\mathbf{x})$, $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_{J_t}]^\top$ where $\phi_i = \pi(\mathbf{x}_i)$ for $i = 1, \ldots, J_t$ Then, we set again $\boldsymbol{\beta} = [\beta_1, ..., \beta_{J_t}] = (\mathbf{K} + \zeta \mathbf{I})^{-1} \boldsymbol{\phi}$ and $\widehat{\phi}_t(\mathbf{x}) = \sum_{i=1}^{J_t} \beta_i k(\mathbf{x}, \mathbf{x}_i)$. The emulator is finally obtained as

$$\widehat{\pi}_t(\mathbf{x}) = \max[\widehat{\phi}_t(\mathbf{x}), 0]. \tag{5.17}$$

Note that these approximations can be directly applied for unbounded support $\mathcal{X}$. We call the scheme based on these constructions as Gaussian Process Adaptive Importance Sampling (GP-AIS).

NN construction. Given $\mathbf{x}_1, \ldots, \mathbf{x}_{J_t} \in \mathcal{X} \subset \mathbb{R}^{d_x}$ (where $\mathcal{X}$ is bounded) and evaluations $\pi(\mathbf{x}_1), \ldots, \pi(\mathbf{x}_{J_t})$, the nearest neighbor (NN) interpolator at $\mathbf{x}$ consists of assigning the value of its nearest node. This is equivalent to consider the Voronoi partition $\mathcal{X} = \cup_{i=1}^{J_t} \mathcal{R}_i$, where

$$\mathcal{R}_i = \{\mathbf{x} \in \mathcal{X} : \|\mathbf{x} - \mathbf{x}_i\| < \left\|\mathbf{x} - \mathbf{x}_j\right\| \text{ for } j \neq i\}, \tag{5.18}$$

is the i-th Voronoi cell. The NN interpolator of π is then given by

$$\widehat{\pi}_t(\mathbf{x}) = \sum_{i=1}^{J_t} \pi(\mathbf{x}_i) \mathbb{I}_{\mathcal{R}_i}(\mathbf{x}), \qquad \mathbf{x} \in \mathcal{X}. \tag{5.19}$$

where $\mathbb{I}_{\mathcal{R}_i}(\mathbf{x})$ is the indicator function in $\mathcal{R}_i$. Note that $\widehat{\pi}$ above is an interpolating approximation of π. The NN search has a cost of $O(J_t)$. We denote the scheme based on this construction as Nearest Neighbor Adaptive Importance Sampling (NN-AIS). The regression case consists in considering the k nearest neighbours to $\mathbf{x}$, and taking the arithmetic mean of the values π in those k nearest nodes.

Remark 8. *Note that RADIS employs an incremental mixture proposal density. Indeed, the emulator $\widehat{\pi}_t(\boldsymbol{x})$ in Eqs.* (5.14)-(5.16) *and* (5.19) *can be expressed as a mixture of pdfs where the number of components, J_t, increases as t grows. For more details of the NN case, see 5.10.2.*

Remark 9. *Under mild conditions, the emulator $\widehat{\pi}_t \to \pi$ and $\widehat{c}_t \to Z$ as $t \to \infty$ (and $L \to \infty$), hence $\frac{1}{\widehat{c}_t}\widehat{\pi}_t \to \bar{\pi}$ (see 5.10.1). Moreover, the SIR scheme to draw from $\frac{1}{\widehat{c}_t}\widehat{\pi}_t$ is asymptotically exact when $L \to \infty$ (see 5.10.1). Hence, RADIS is drawing samples from $\bar{\pi}$, i.e., it is asymptotically an exact sampler.*

The GP construction provides smoother solutions that can be directly employed in unbounded domains. However, the GP requires the inversion of matrix (with a dimension that increases as the number of nodes grows) and the tuning of the hyperparameters of the kernel function. In contrast, the NN construction does not need any matrix inversion and, if we fix in advance the number k neighbours (for instance in the interpolation case, we have $k = 1$) no hyperparameter tuning is required.

5.5. Robust accelerating schemes

In this section, we present some alternatives in order to **(a)** reduce the dependence from the initial nodes and **(b)** increase the applicability of RADIS, **(c)** speed up the convergence of the emulator covering quickly the state space and finally **(d)** we discuss the computational cost of the proposed overall scheme. The resulting methods are robust schemes, which can be also employed for extending the use of NN-AIS in unbounded supports. This is achieved combining the non-parametric proposal function $\widehat{\pi}_t(\mathbf{x})$ with a parametric proposal density, $q_{\text{par}}(\mathbf{x})$. Hence, the complete proposal, denoted as $\varphi_t(\mathbf{x})$. will be a mixture of densities with a parametric and a non-parametric components.

Mixture with parametric proposal. The use of an additional parametric density $\bar{q}_{\text{par}}(\mathbf{x})$ can (i) ensure that the complete proposal has fatter tails than target pdf, and (ii) foster the exploration of important regions that could be initially ignored due to a possible bad initialization. Thus, we consider the following mixture as a proposal density in the inner IS layer,

$$\varphi_t(\mathbf{x}) = \alpha_t \bar{q}_{\text{par}}(\mathbf{x}) + (1 - \alpha_t)\frac{1}{\hat{c}_t}\widehat{\pi}_t(\mathbf{x}), \tag{5.20}$$

where $\alpha_t \in [0, 1]$ for all t, and α_t is a non-increasing function t. The idea is to set initially $\alpha_0 = \frac{1}{2}$, and then decrease $\alpha_t \to \alpha_\infty$ as $t \to \infty$ (e.g., we can set $\alpha_\infty = 0$). Note that $\varphi_t(\mathbf{x})$ must be evaluated in the denominator of the outer layer weights $w_{t,n}$ in (5.11), taking the place of $\frac{1}{\widehat{c}_t}\widehat{\pi}_t(\mathbf{x})$ (see Table 5.1).

Remark 10. *Choosing $\bar{q}_{\text{par}}(\boldsymbol{x})$ with fatter tails than $\bar{\pi}(x)$, then $\varphi_t(\boldsymbol{x})$ has also fatter tails than $\bar{\pi}(x)$. Hence, we avoid the infinite variance issue of the IS weights [71].*

See also [39, Section 7.1] for a theoretical and numerical example of the infinite variance problem. As an example, if $\mathcal{X}$ is bounded, $\bar{q}_{\text{par}}(\mathbf{x})$ could be a uniform density over $\mathcal{X}$. If $\mathcal{X}$ is unbounded, $\bar{q}_{\text{par}}(\mathbf{x})$ can be, e.g., a Gaussian, a Student-t distribution or a mixture of pdfs (see below).

Remark 11. *The fact that $\varphi_t(\boldsymbol{x})$ has fatter tails than $\bar{\pi}(x)$ ensures to have a non-zero probability of adding new nodes in any possible subset of the support $\mathcal{X}$.*

This strategy also allows the use of the NN-AIS in an unbounded support. In 5.10.3 we describe an extension of NN-AIS where the support of the NN approximation is also adapted.

Parametric mixture by other AIS schemes. A more sophisticated option is to also update $\bar{q}_{\text{par}}(\mathbf{x})$ along the iterations. For instance, $\bar{q}_{\text{par}}(\mathbf{x}) = \frac{1}{C}\sum_{c=1}^{C} q_c(\mathbf{x}|\boldsymbol{\mu}_{t,c}, \boldsymbol{\Sigma}_{t,c})$ can be itself a mixture, whose parameters are adapted following another AIS scheme, so that the

complete proposal would be

$$\varphi_t(\mathbf{x}) = \alpha_t \left(\frac{1}{C} \sum_{c=1}^{C} \bar{q}_c(\mathbf{x}|\boldsymbol{\mu}_{t,c}, \boldsymbol{\Sigma}_{t,c}) \right) + (1 - \alpha_t) \frac{1}{c_t} \widehat{\pi}_t(\mathbf{x}), \tag{5.21}$$

with $\alpha_t \in [0, 1]$ for all t. As an example, the parametric mixture $\bar{q}_{\texttt{par}}(\mathbf{x})$ can be obtained following a population Monte Carlo (PMC) method, or a layered adaptive importance sampling (LAIS) technique and/or adaptive multiple importance sampling (AMIS) scheme [4]. The weight α_t is again a non-increasing function of the iteration t.

Regression versus interpolation. In the first iterations of RADIS, the use of $\zeta > 0$ in the GP approximation and/or considering the k nearest neighbours (instead only the closest one, $k = 1$), also decreases the dependence on the initial nodes. Namely, reducing the overfitting, at least in the first iteration of RADIS, also increases the robustness of the algorithm.

More layers. To leverage the benefits of different emulator constructions in RADIS, one possible strategy is to employ additional layers in the deep architecture, as depicted in Fig. 5.3. For instance, with one additional layer, we could use jointly the GP and the NN constructions. Another possibility is to consider several GP models with different kernel functions or several NN schemes with different k.

5.5.1. Computational cost

In this section, we discuss computational details of our approach and hypothesize when our approach is convenient also in terms of computational time. It is important to remark that RADIS is useful also for constructing a good emulator (not just for approximating integrals as other Monte Carlo schemes), choosing the nodes in a proper way, similarly in an active learning scheme [40, 79]. Figures 5.6(d) and 5.11 in the numerical experiments provide a comparison with a random addition of nodes, showing the benefits of the adaptive construction employed in RADIS.

RADIS requires N evaluations of the posterior $\pi(\mathbf{x})$ at each iteration, so that the total number of posterior evaluations is $E = N_0 + NT$. Let denote as $C_{\text{eval-post}}$ the cost of evaluating $\pi(\mathbf{x})$ once, so that the total cost of evaluating the posterior is $EC_{\text{eval-post}}$. In addition to E posterior evaluations, RADIS carries out different other tasks, namely (i) evaluate L times the current emulator per iteration, (ii) perform N resampling steps per iteration over L possible samples, and (iii) compute the denominator of thefinal IS weights at the end of the algorithm. Let $C_{\text{eval-emulator}}$, $C_{\text{resampling}}$ and $C_{\text{den-weights}}$ denote the *total* costs after T iterations of RADIS, associated to tasks (i)-(iii). In term of computational time, RADIS can be convenient with respect to other schemes, when the inequality

$$C_{\text{eval-post}} > \frac{1}{E} \left(C_{\text{eval-emulator}} + C_{\text{resampling}} + C_{\text{den-weights}} \right), \tag{5.22}$$

is fulfilled. For an example, see the numerical experiment in Section 5.8.3 and the results in Table 5.9. Recall that all the values $C_{\text{eval-post}}$, $C_{\text{eval-emulator}}$, $C_{\text{resampling}}$, and $C_{\text{den-weights}}$

also depend on the specific implementation and language of the code and the different processors/machines.
Generally, the term $C_{\text{eval-emulator}}$ dominates the other two since it is composed of evaluating L times the emulator for T iterations. Moreover, due to the non-parametric construction and the fact that we increase the set of active nodes in N, evaluating the interpolator becomes more costly with the iterations. More specifically, in the NN based approach, after T iterations we have $C_{\text{eval-emulator}} \approx \sum_{t=1}^{T} O(LNt) = O(LNT^2)$. In the GP-AIS scheme, we have the additional cost of inverting the $J_t \times J_t$ matrix at each iteration (recall that $J_t = N_0 + N(t-1)$). This cost at each iteration is $O(J_t^3) \approx O(N^3 t^3)$, for t big enough. Then, in GP-AIS, $C_{\text{eval-emulator}} \approx \sum_{t=1}^{T} O(N^3 t^3) + O(LNT^2) = O(N^3 T^4) + O(LNT^2)$.
In the next section, we describe different procedures to decrease $C_{\text{eval-emulator}}$.

5.6. Construction of parsimonious emulators

So far, we have considered updating the interpolant at each iteration t by adding all the N samples drawn at that iteration. In order to control the computational cost of evaluating the emulator, we can design a strategy for accepting or rejecting some of the possible additional nodes. This can be done assigning acceptance probabilities, $p_{\text{A}}(\mathbf{x}_{t,n}) \in [0,1]$, to each of the N samples (in the same fashion of [44, 43]). Therefore, the update part of Step 3 in Table 5.1 would be replaced by the routine in Table 5.2.

Table 5.2: **Parsimonious update in Step 3 of Table 5.1.**

- **Initialization:** Choose an acceptance function $p_A(\mathbf{x})$, set $\mathcal{S}_t = \mathcal{S}_{t-1}$, and consider the cloud of resampled particles $\{\mathbf{x}_{t,n}\}_{n=1}^{N}$, from the previous step of Table 5.1.
- For $n = 1, \ldots, N$**:**

1. Draw $u \sim \mathcal{U}([0,1])$.
2. If $u \leq p_A(\mathbf{x}_{t,n})$, then set $\mathcal{S}_t = \mathcal{S}_t \cup \{\mathbf{x}_{t,n}\}$. Otherwise, If $u > p_A(\mathbf{x}_{t,n})$, discard $\mathbf{x}_{t,n}$.

-**Output:** Return $\mathcal{S}_t$ and $J_t = |\mathcal{S}_t|$.

Proper acceptance functions. We say that an acceptance probability, $p_{\text{A}}(\mathbf{x}) : \mathcal{X} \to [0,1]$, is *proper* if satisfies

$$\textbf{C1:} \quad p_{\text{A}}(\mathbf{x}) \to 0, \quad \text{if} \quad |\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})| \to 0, \tag{5.23}$$

for any $\mathbf{x} \in \mathcal{X}$, and

$$\textbf{C2:} \quad p_{\text{A}}(\mathbf{x}) = 0 \text{ if and only if } |\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})| = 0. \tag{5.24}$$

Hence, for any node contained already in $\mathcal{S}_{t-1}$, i.e., $\mathbf{z} \in \mathcal{S}_{t-1}$, we have $p_{\text{A}}(\mathbf{z}) = 0$. For this reason, as we show below, the acceptance function often depends on the current emulator

$\widehat{\pi}_t(\mathbf{x})$, i.e., we should write $p_A(\mathbf{x}) = p_A(\mathbf{x}|\widehat{\pi}_t)$. Hence, a more precise and parsimonious construction would consider a sequential updating of the emulator (since $p_A(\mathbf{x})$ also should change during the acceptance tests), as shown in Table 5.3.

Table 5.3: **Alternative parsimonious update considering a sequential updating of the emulator.**

- **Initialization:** Set $\widehat{\pi}_t^{(0)}(\mathbf{x}) = \widehat{\pi}_t(\mathbf{x})$, choose an acceptance function $p_A^{(0)}(\mathbf{x}) = p_A(\mathbf{x}|\widehat{\pi}_t^{(0)})$, set $k = 0$ and $\mathcal{S}_t = \mathcal{S}_{t-1}$, and consider the cloud of resampled particles $\{\mathbf{x}_{t,n}\}_{n=1}^N$, from the previous step of Table 5.1. Note that, more generally, $p_A^{(k)}(\mathbf{x}) = p_A(\mathbf{x}|\widehat{\pi}_t^{(k)})$ where $k \geq 0$ is an index.
- **For** $n = 1, \ldots, N$**:**

1. Draw $u \sim \mathcal{U}([0,1])$.
2. If $u \leq p_A^{(k)}(\mathbf{x}_{t,n})$, then set $\mathcal{S}_t = \mathcal{S}_t \cup \{\mathbf{x}_{t,n}\}$, and update the emulator construction $\widehat{\pi}_t^{(k+1)}(\mathbf{x})$ considering the new set $\mathcal{S}_t$. Set also $k \leftarrow k+1$.

-**Output:** Return $\mathcal{S}_t$, $J_t = |\mathcal{S}_t|$ and $\widehat{\pi}_{t+1}(\mathbf{x}) = \widehat{\pi}_{t+1}(\mathbf{x}; \mathcal{S}_t) = \widehat{\pi}_t^{(k)}(\mathbf{x})$.

Remark 12. *Note that the procedures in Table 5.2 and 5.3 do not require additional evaluations of the target π, since all the values $\pi(\boldsymbol{x}_{t,n})$, for all n, are already obtained.*

The difference between the schemes in Tables 5.2 and 5.3, in term of performance and computational cost, becomes more relevant as N grows. Note that the order of the tests in Table 5.3 could be also relevant and some strategies for ordering $\{\mathbf{x}_{t,n}\}_{n=1}^N$ (in a suitable way) could be designed. Below, we introduce some examples of proper acceptance functions and also some reasonable improper ones.

5.6.1. Examples of proper acceptance functions

One possibility of proper acceptance function is

$$\textbf{A1:} \quad p_{\mathrm{A}}(\mathbf{x}) = 1 - \frac{\min\{\pi(\mathbf{x}), \widehat{\pi}_t(\mathbf{x})\}}{\max\{\pi(\mathbf{x}), \widehat{\pi}_t(\mathbf{x})\}} = \frac{|\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})|}{\max\{\pi(\mathbf{x}), \widehat{\pi}_t(\mathbf{x})\}}, \tag{5.25}$$

where we have used $|\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})| = \max\{\pi(\mathbf{x}), \widehat{\pi}_t(\mathbf{x})\} - \min\{\pi(\mathbf{x}), \widehat{\pi}_t(\mathbf{x})\}$. Another possibility is to consider both the discrepancy between π and $\widehat{\pi}_t$, and the distance to the closest node $\mathbf{s}^* \in \mathcal{S}_{t-1}$ to $\mathbf{x}$, i.e.,

$$\textbf{A2:} \quad p_{\mathrm{A}}(\mathbf{x}) = \left(1 - e^{-\alpha|\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})|}\right)\left(1 - e^{-\beta\|\mathbf{x} - \mathbf{s}^*\|}\right), \quad \alpha, \beta \geq 0. \tag{5.26}$$

If either $\alpha = 0$ or $\beta = 0$ (or both), then $p_{\mathrm{A}}(\mathbf{x}) = 0$. As $\alpha \to \infty$ and $\beta \to \infty$ grow, then $p_{\mathrm{A}}(\mathbf{x}) \to 1$. When $\alpha = \infty$ and β is finite, then $p_{\mathrm{A}}(\mathbf{x}) = 1 - e^{-\beta\|\mathbf{x} - \mathbf{s}^*\|}$ and the acceptance probability is bigger when the point $\mathbf{x}$ is far from its closest node, i.e., we have a

space-filling strategy. When α is finite and $\beta \to \infty$, then $p_{\mathrm{A}}(\mathbf{x}) = 1 - e^{-\alpha|\pi(\mathbf{x})-\widehat{\pi}_t(\mathbf{x})|}$, and the acceptance probability is bigger if there is a large discrepancy between π and the interpolant at $\mathbf{x}$. Thus, unlike in (5.25), in (5.26) we should tune the values α, and β according to the computational budget we have, or according to the trade-off between computational effort and performance.

Note that, in the acceptance functions above, we have $p_{\mathrm{A}}(\mathbf{x}) \in [0, 1]$ for all $\mathbf{x}$, and the condition (5.23) is fulfilled. Moreover, these acceptance functions depend only on $\mathbf{x}$, $\pi(\mathbf{x})$ and $\widehat{\pi}_t(\mathbf{x})$. The decision is done considering the quality of the approximation of $\widehat{\pi}_t(\mathbf{x})$ and, in Eq. (5.26), the relative position of $\mathbf{x}$ with respect to the nodes in $\mathcal{S}_{t-1}$. They do not depend on the rest of $N-1$ possible nodes within $\{\mathbf{x}_{t,n}\}_{n=1}^N$ to be tested. Nevertheless, if we use the sequential updating scheme of Table 5.3, the acceptance probability will change depending on the order in which we test the candidate nodes.

An example of proper acceptance function depending on the population of candidate nodes is described next. Let us define $R(\mathbf{x}) = |\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})|$. Considering $\mathbf{x} \in \{\mathbf{x}_{t,1}, ..., \mathbf{x}_{t,N}\}$ (i.e., one point within the set of possible nodes to be included) and defining $R_{\mathtt{max}} = \max_{\mathbf{x}\in\{\mathbf{x}_{t,1},...,\mathbf{x}_{t,N}\}} R(\mathbf{x})$, we can set

$$\textbf{A3:} \quad p_{\mathrm{A}}(\mathbf{x}) = \frac{R(\mathbf{x})}{R_{\mathtt{max}}}, \quad \text{with} \quad \mathbf{x} \in \{\mathbf{x}_{t,n}\}_{n=1}^N. \tag{5.27}$$

Again $p_{\mathrm{A}}(\mathbf{x}) \in [0, 1]$ and the condition (5.23) is satisfied. Note that a normalization of $R(\mathbf{x})$ using $\sum_{n=1}^N R(\mathbf{x}_{t,n})$ instead of $R_{\mathtt{max}}$ would produce very small acceptance probabilities as N grows (note that $R(\mathbf{x}) \geq 0$ for all $\mathbf{x}$). This is a non beneficial effect in our opinion, since the decrease of $p_{\mathrm{A}}(\mathbf{x})$ is not due to a good quality of the approximation $\widehat{\pi}_t$, but is generated by the increase of the possible alternative denominator $\sum_{n=1}^N R(\mathbf{x}_{t,n})$. Resampling schemes could be also employed but provide improper acceptance functions, as we discuss below.

5.6.2. Examples of improper acceptance functions

Let us define the auxiliary weights $\rho(\mathbf{x}) = \frac{F(\mathbf{x})}{\widehat{\pi}_t(\mathbf{x})}$ where $F(\mathbf{x})$ is function that can chosen in different ways, $F(\mathbf{x}) = \pi(\mathbf{x})$, $F(\mathbf{x}) = |\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})|$ or $F(\mathbf{x}) = |\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})|\widehat{\pi}_t(\mathbf{x})$, for instance. The nodes to be included are then selected resampling N times within the set $\{\mathbf{x}_{t,n}\}_{n=1}^N$ according to the following probability mass,

$$\bar{\rho}(\mathbf{x}_{t,i}) = \frac{\rho(\mathbf{x}_{t,i})}{\sum_{n=1}^N \rho(\mathbf{x}_{t,n})}, \quad i = 1, ..., N,$$

and taking only the *unique* values (i.e., without repetitions). Table 5.4 summarizes this idea.

The acceptance probability is, in this case,

$$p_A(\mathbf{x}_{t,i}) = 1 - (1 - \bar{\rho}(\mathbf{x}_{t,i}))^N. \tag{5.28}$$

Thus, the procedure in Table 5.4 is equivalent (in term of number of added nodes) to apply the procedure in Table 5.2 and $p_A(\mathbf{x})$ in (5.28) above. Observe also that, with these

Table 5.4: **Parsimonious update in Step 3 of Table 5.1 based on resampling.**

- **Initialization:** Choose a numerator function $F(\mathbf{x})$ (e.g., $F(\mathbf{x}) = \pi(\mathbf{x})$ or $F(\mathbf{x}) = |\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})|$) for the weight $\rho(\mathbf{x}) = \frac{F(\mathbf{x})}{\widehat{\pi}_t(\mathbf{x})}$. Set $\mathcal{S}_t = \mathcal{S}_{t-1}$, and consider the cloud of resampled particles $\{\mathbf{x}_{t,n}\}_{n=1}^N$, from the previous step of Table 5.1. Then:

1. Resample N times within $\{\mathbf{x}_{t,n}\}_{n=1}^N$ according to the probability mass defined as

$$\bar{\rho}_{t,i} = \bar{\rho}(\mathbf{x}_{t,i}) = \frac{\rho(\mathbf{x}_{t,i})}{\sum_{n=1}^N \rho(\mathbf{x}_{t,n})}, \quad i = 1, ..., N,$$

   obtaining the new set $\{\widetilde{\mathbf{x}}_{t,n}\}_{n=1}^N$.

2. Take the unique values in $\{\widetilde{\mathbf{x}}_{t,n}\}_{n=1}^N$ (i.e., removing the repetitions) obtaining $\{\mathbf{v}_{t,k}\}_{k=1}^K$ (where K is the number of unique values in $\{\widetilde{\mathbf{x}}_{t,n}\}_{n=1}^N$).

3. Set $\mathcal{S}_t = \mathcal{S}_t \cup \{\mathbf{v}_{t,1}, ..., \mathbf{v}_{t,K}\}$.

-**Output:** Return $\mathcal{S}_t$ and $J_t = |\mathcal{S}_t|$.

schemes, even in the ideal case $\widehat{\pi}_t(\mathbf{x}) = \pi(\mathbf{x})$ for all $\mathbf{x}$, we always add at least one node to the new sets $\mathcal{S}_t$ (i.e., $K \geq 1$). This is due to the *improperness* of the acceptance functions. Then, these resampling-based schemes could possibly yield less parsimonious emulators. Nevertheless, they are easy to implement and their implementation is computationally faster than the rest of approaches, described previously. Starting from the samples $\mathbf{z}_{t,\ell} \sim q_{\text{aux}}(\mathbf{x})$ in RADIS, the added points $\{\mathbf{v}_{t,k}\}_{k=1}^K$ in Table 5.4 are then obtained as results of two resampling procedures and finally considering the unique values:

$$\{\mathbf{z}_{t,\ell}\}_{\ell=1}^L \xrightarrow{\bar{\gamma}_{t,\ell}} \{\mathbf{x}_{t,n}\}_{n=1}^N \xrightarrow{\bar{\rho}_{t,\ell}} \{\mathbf{v}_{t,k}\}_{k=1}^K.$$

In the vanilla version of RADIS, the nodes are obtained applying just the first resampling at each iteration. Another example of *improper* acceptance function that is not based on a resampling procedure (and does not take into account all the population $\{\mathbf{x}_{t,n}\}_{n=1}^N$, jointly) is

$$p_{\text{A}}(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad |\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})| > \epsilon, \\ 0 & \text{if} \quad |\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})| \leq \epsilon, \end{cases} \quad \text{for} \quad \epsilon \geq 0. \tag{5.29}$$

Note that for a finite positive value of $\epsilon > 0$, after some iterations, possibly we will have $p_{\text{A}}(\mathbf{x}) = 0$, i.e., the adaptation of the emulator is stopped. This is the reason of its improperness, since it does not fulfill C2. If $\epsilon = 0$, then we always have $p_{\text{A}}(\mathbf{x}) = 1$, adding all the nodes. If $\epsilon = \infty$, we have always $p_{\text{A}}(\mathbf{x}) = 0$, and we never update the emulator. With a suitable choice of ϵ (tuned according to computational budget available), this acceptance function can be also a good option. A numerical comparison among these acceptance probabilities is given in Section 5.8.

5.7. RADIS for model emulation and sequential inversion

In this section, we describe the application of RADIS to solve Bayesian inverse problems. We have already considered the case of obtaining a surrogate function for the (unnormalized) density π (or $\log \pi$). We here focus on inverse inference problems where our aim is also to obtain an emulator of the costly forward model. More specifically, let us consider a generic Bayesian inversion problem

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v}. \tag{5.30}$$

where $\mathbf{h}(\mathbf{x}) : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ represents a non-linear mapping defining a physical or mechanistic model (e.g. a complex energy transfer model, a climate model subcomponent integrating subgrid physical processes, or a set of differential equations describing a chemical diffusion process) and $\mathbf{v}$ has a multivariate Gaussian pdf (e.g., with zero mean and a diagonal covariance matrix with σ^2 in the diagonal). Considering a prior $g(\mathbf{x})$ over $\mathbf{x}$, the posterior is

$$\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x}) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{h}(\mathbf{x})\|^2\right) g(\mathbf{x}),$$

which can be costly to evaluate if $\mathbf{h}(\mathbf{x})$ is a complex model. In this setting, it is often required to build an emulator of the physical model $\mathbf{h}(\mathbf{x})$ instead of a surrogate function for the pdf π [64, 35, 7, 78]. However, we can build $\widehat{\mathbf{h}}(\mathbf{x})$ using the same procedures in Sect. 5.4.2, and then obtain

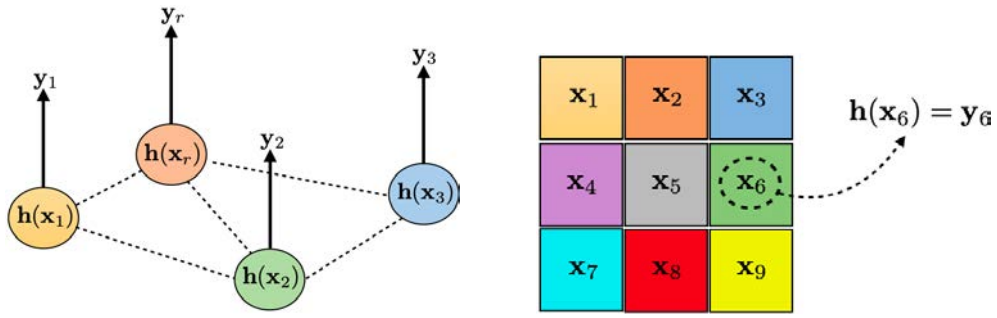
$$\widehat{\pi}(\mathbf{x}) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y} - \widehat{\mathbf{h}}(\mathbf{x})\|^2\right) g(\mathbf{x}),$$

which can be employed as proposal in our scheme. Hence, in this case, we obtain two emulators: $\widehat{\mathbf{h}}(\mathbf{x})$ of the physical model, and $\widehat{\pi}(\mathbf{x})$ of the posterior.

In many real-world applications, we have a sequence of inverse problems

$$\mathbf{y}_r = \mathbf{h}(\mathbf{x}_r) + \mathbf{v}_r, \quad r = 1, \ldots, R, \tag{5.31}$$

where R denotes the number of observation nodes in the network, but the physical model $\mathbf{h}$ is the same for all nodes. See an illustrative example in Fig. 5.4(a). The underlying graph represents different features and may have different statistical meanings. Moreover, it can contain prior information directly given in the specific problem. As an example, consider the case of an image where each pixel is represented as a node in the network, see Fig. 5.4(b), and the goal is to retrieve a set of parameters $\mathbf{x}$ from the observed or simulated pixels $\mathbf{y}$. This is the standard scenario in remote sensing applications, where the observations $\mathbf{y}$ are very high dimensional (depending on the sensory system and satellite platform ranging from a few spectral channels to even thousands) and the set of parameters $\mathbf{x}$ describe the physical characteristics of each particular observation (e.g. leaf or canopy structure, observation characteristics, vegetation health and status, etc). In other settings the graph must be also inferred, i.e., the connections should be learned as well. A



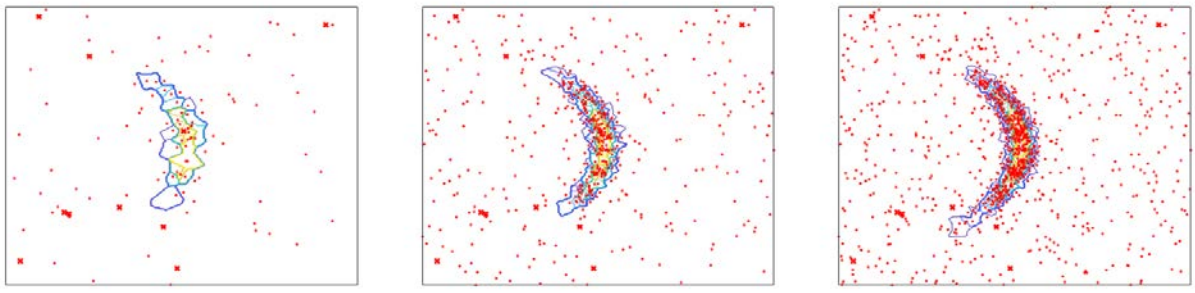
(a) (b)

Figure 5.4: **(a)** Different inversion problems related to each other involving the same underlying physical model $\mathbf{h}(\cdot)$. Their relationships are represented by (dashed lines) edges between the nodes. **(b)** Example of network in an image, where each pixel represents a node of the network. This is the scenario in remote sensing image processing, where $\mathbf{x}_i$ represents the physical state parameters to infer from a set of acquired (or simulated) spectra $\mathbf{y}_i$ (in this figure, we consider noise-free observations).

simple strategy is to consider the strength of the link is proportional to $\exp\left(-\|\mathbf{y}_r - \mathbf{y}_j\|\right)$, for instance. Other more sophisticated procedures can be also employed [16]. Given Eq. (5.31), a piece of the likelihood function is

$$p(\mathbf{y}_r|\mathbf{x}_r) \propto \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y}_r - \mathbf{h}(\mathbf{x}_r)\|^2\right), \quad r = 1, \ldots, R.$$

Note that the observation model $\mathbf{h}(\cdot)$ is shared in all the R nodes. The complete likelihood function is $p(\mathbf{y}_{1:r}|\mathbf{x}_{1:r}) = p(\mathbf{y}_1, ..., \mathbf{y}_R|\mathbf{x}_1, ..., \mathbf{x}_R) = \prod_{i=1}^{R} p(\mathbf{y}_r|\mathbf{x}_r)$. A complete Bayesian analysis can be considered in this scenario, implementing also RADIS within a particle filter for an efficient inference. However, it is out of the scope of this work and we leave it as a future research line.



(a) (b) (c)

Figure 5.5: Evolution of $\widehat{\pi}_t$ from NN-AIS+U through iterations **(a)** $t = 10$, **(b)** $t = 50$, and **(c)** $t = 100$.

5.8. Numerical experiments

In this section, we provide several numerical tests in order to show the performance of the proposed scheme and compare them with benchmark approaches in the literature. The first example corresponds to a nonlinear banana shaped density in dimension $d_x = 2$, where we compare NN-AIS against standard IS algorithms. The second test is a multimodal scenario with dimension $d_x = 10$, where we test the combination of an AIS algorithm with NN-AIS against other AIS. An application to an astronomical model is also given, where we provide a comparison in terms of computation time. Finally, we consider an application to remote sensing, specifically, we test our scheme in multiple bayesian inversions of PROSAIL.

5.8.1. Toy example 1: banana-shaped density

We consider a banana shaped target pdf,

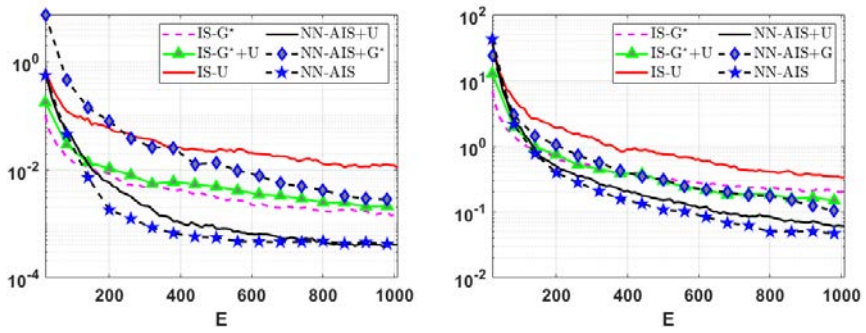
$$\bar{\pi}(\mathbf{x}) \propto \exp\left(-\frac{(\eta_1 - Bx_1 - x_2^2)^2}{2\eta_0^2} - \sum_{i=1}^{d_x} \frac{x_i^2}{2\eta_i^2}\right), \tag{5.32}$$

with $B = 4$, $\eta_0 = 4$ and $\eta_i = 3.5$ for $i = 1, ..., d_x$, where $\mathcal{X} = [-10, 10] \times [-10, 10]$, i.e., bounded domain. We consider $d_x = 2$ and compute in advance Z and the mean of the target (i.e., the groundtruth) by using a costly grid, so that we can check the performance of the different techniques.

Estimating Z and μ

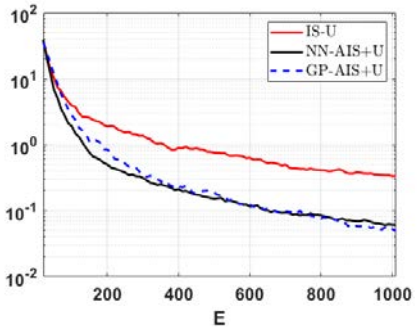
We aim to estimate $Z = 7.9976$ and $\boldsymbol{\mu} = [-0.4841, 0]$ with NN-AIS and compare it, in terms of relative mean squared error (RMSE), with different IS algorithms considering the same number of target evaluations. The results are averaged over 500 independent simulations. The goal is to investigate the performance of NN-AIS as compared to other parametric IS algorithms that consider a proposal, well designed in advance. We set $T = 100$ and $N = 10$, and use 10 starting nodes (random chosen in the domain) to build $\widehat{\pi}_1(\mathbf{x}|\mathcal{S}_0)$. With the selected values of T and N the total budget of target evaluations is $E = 10 + NT = 1010$.

Methods. We consider three variants of NN-AIS to illustrate three different scenarios: in the first one (denoted as **NN-AIS**) initial nodes uniform in $[-10, 10] \times [-10, 10]$, i.e. good initialization, without $\bar{q}_{\text{par}}(\mathbf{x})$; (**NN-AIS+U**) same initialization with $\bar{q}_{\text{par}}(\mathbf{x}) = \frac{1}{|\mathcal{X}|}$, i.e. good initialization and with a good choice of $\bar{q}_{\text{par}}(\mathbf{x})$; (**NN-AIS+G**) initial nodes are uniform in $[5, 10] \times [5, 10]$ with Gaussian $\bar{q}_{\text{par}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|[2, 2]^\top, 3^2\mathbf{I}_2)$, i.e., a bad initialization with a bad choice of the parametric proposal $\bar{q}_{\text{par}}(\mathbf{x})$. In all cases, we consider a fixed value of $\alpha_t = \frac{1}{2}$.

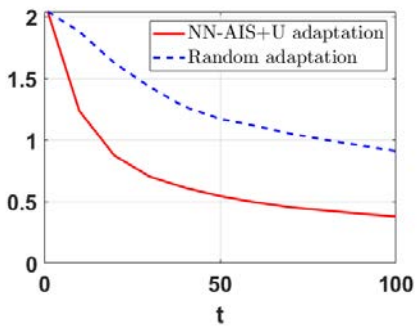


(a) RMSE for Z

(b) RMSE for $\boldsymbol{\mu}$



(c) RMSE for $\boldsymbol{\mu}$

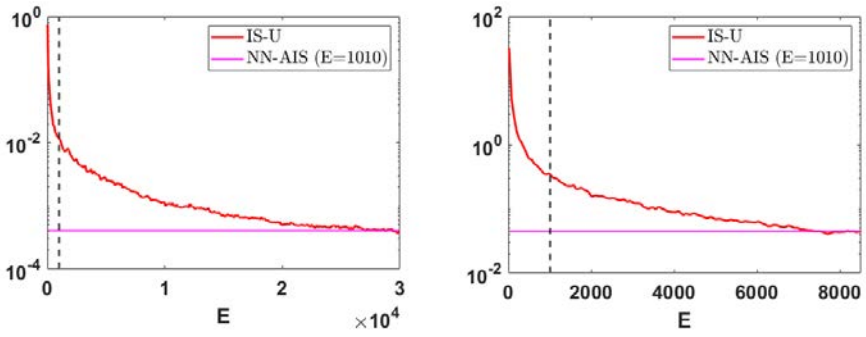


(d) L_2 distance between $\widehat{\pi}_t$ and π versus t

Figure 5.6: **(a)** RMSE in log-scale for Z as function of evaluations E. **(b)** RMSE in log-scale for $\boldsymbol{\mu}$ as function of E. **(c)** RMSE of GP-AIS+U in log-scale for $\boldsymbol{\mu}$ as function of E. **(d)** L_2 distance between π and $\widehat{\pi}_t$ when the nodes are adaptively obtained by NN-AIS+U (in solid line), and when the nodes are random and uniformly chosen in the domain (in dashed line), as a function of t.

Furthermore, we compare the NN-AIS schemes with three alternative IS methods: (**IS-U**) with uniform proposal in $\mathcal{X}$, which is very good choice of proposal in this problem; (**IS-G***) with Gaussian proposal matching the moments of $\bar{\pi}(\mathbf{x})$, i.e., the optimal Gaussian proposal; (**IS-G*+U**) with a proposal which is an equally weighted mixture of the two previous cases. In addition, we also test our algorithm using GPs, denoted GP-AIS+U.

Discussion. As shown in Figures 5.6(a)-(b), NN-AIS and NN-AIS+U outperform the rest. NN-AIS performs a bit better than NN-AIS+U: the use of a parametric proposal is safer but entails a loss of performance, trading off exploitation for exploration. In Figure 5.6(a), NN-AIS+G shows worse performance in estimating Z in the early iterations as a consequence of the bad initialization and bad parametric proposal. However, it quickly improves and start performing as good as IS-G* and IS-G*+U. In Figure 5.6(b), regarding the estimation of $\boldsymbol{\mu}$, our methods perform better than alternative IS algorithms. Figure 5.6(c) shows that GP-AIS+U provides similar performance than NN-AIS+U. Overall, this simple experiment shows the range of performance of our method: it is best if we use only our method, provided that we have a good initialization; adding a good parametric proposal is safer if we do not trust our initialization, showing just a small loss of performance w.r.t. the first scenario. In the case both the initialization and parametric proposal



(a) (b)

Figure 5.7: We show the number of additional evaluations required by IS-U to achieve the same RMSE than NN-AIS with $E = 1010$ in **(a)** the estimation of Z, and **(b)** the estimation of $\boldsymbol{\mu}$. The red line represents the RMSE of IS-U as a function of E, while the horizontal line is the RMSE achieved by NN-AIS with $E = 1010$. The vertical dash line is at $E = 1010$.

are wrongly chosen, our method is able to achieve good results and recover quickly from a bad initialization.

Additional comparison. we have run IS-U for $E > 1010$ until it reached the same error in estimation achieved by NN-AIS. The results are depicted in Figures 5.7. Specifically, in Figure 5.7(a) we see that around 29000 more evaluations are needed to obtain the same error in estimating Z, and Figure 5.7(b) shows that around 7000 more evaluations to obtain the same error in estimating $\boldsymbol{\mu}$.

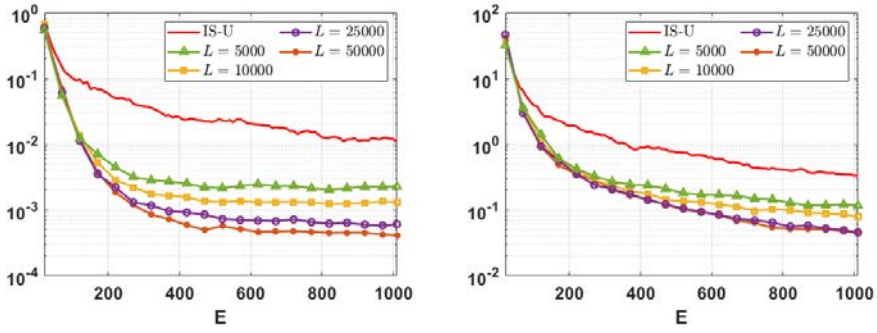
Convergence of $\widehat{\pi}_t$ to π

The convergence of $\widehat{\pi}_t$ to π depends on the fact that nodes should fill the space enough (see 5.10.1). However, some filling strategies yield a faster convergence than others. In our simulations, we aim to show that the construction provided by NN-AIS+U converges faster than another construction using nodes random and uniformly chosen in the domain $\mathcal{X}$. Figures 5.5 and 5.6(d) show that the approximation $\widehat{\pi}_t$ obtained by NN-AIS+U is indeed converging to π as t increases. In Figure 5.6(c), we show the L_2 distance between π and $\widehat{\pi}_t$ with random nodes (in dashed line), and by NN-AIS+U (in solid line), along with the number of iterations t. As shown in Figure 5.6(d), the $\widehat{\pi}_t$ gets more rapidly closer to π in L_2 when the nodes are sampled from NN-AIS+U rather than only adding random points, uniformly over the domain.

Comparing NN-AIS+U with different values of L

In our proposed approach, we need to evaluate L times the approximation $\widehat{\pi}_t$ at each iteration. The computation cost of the algorithm thus scales with L, which needs to be big enough (and bigger than N) so that the resampling step and the estimation of c_t are

accurate. Here, we investigate the performance of NN-AIS+U for several values $L \in \{5000, 10000, 25000, 50000\}$. As expected, Figure 5.8 shows that the performance of the algorithm deteriorates as we lower the value of L. However, note that all NN-AIS scheme with the considered L perform better compared to standard IS with uniform proposal.



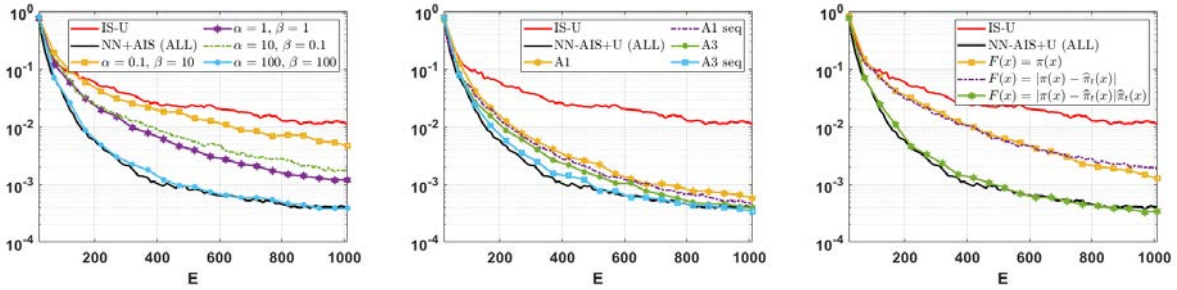
(a) (b)

Figure 5.8: Performance of NN-AIS+U with different choices of $L \in \{1000, 5000, 10000, 25000, 50000\}$ in **(a)** the estimation of Z, and **(b)** the estimation of $\boldsymbol{\mu}$. The red curve represents the RMSE of IS-U as a function of E.

Results of the parsimonious constructions

In the vanilla version of RADIS, the approximation $\widehat{\pi}_t$ is refined by adding the N samples drawn at iteration t to the set of active nodes. Since we consider non-parametric approximations, this implies that $\widehat{\pi}_t$ becomes more complex, i.e. more costly to evaluate, as t grows. In Sect. 5.6, we showed means of controlling the complexity of $\widehat{\pi}_t$ by the computation of acceptance probabilities: instead of adding all the samples, the n-th sample is added with certain probability. Here, we test the application of several acceptance probabilities to NN-AIS+U and compare the performance with respect to NN-AIS+U that accepts all nodes. We also examine the complexity, in terms of number of nodes, of the final emulator. Specifically, we consider the acceptance functions **A1** in Eq. (5.25), **A2** in Eq. (5.26) and **A3** in Eq. (5.27). We also test three variants of the improper acceptance function in Sect. 5.6.2, namely $F(\mathbf{x}) = \pi(\mathbf{x})$, $F(\mathbf{x}) = |\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})|$ and $F(\mathbf{x}) = |\pi(\mathbf{x}) - \widehat{\pi}_t(\mathbf{x})|\widehat{\pi}_t(\mathbf{x})$. The results are given in Figures 5.9, Figure 5.10 and Figure 5.11. Note that NN-AIS+U (ALL) represents the vanilla version NN-AIS+U in Table 5.1, adding all the nodes at the Step 3.

Figure 5.9(a) shows the application of the acceptance probability **A2** for different choices of α and β using the updating scheme in Table 5.3. Recall that, when α or β are 0, the acceptance probability is 0. When $\alpha \gg 1$ and $\beta > 0$, the nodes are added in a space-filling fashion. On the contrary, when $\beta \gg 1$ and $\alpha > 0$, the nodes are added by accounting for the discrepancy between π and $\widehat{\pi}_t$. We note that the former strategy works better than the latter, as shown in Figure 5.9(a). Moreover, the performance is better when $\alpha = \beta = 1$, that is, both strategies at the same time. As α and β grow, we recover the per-

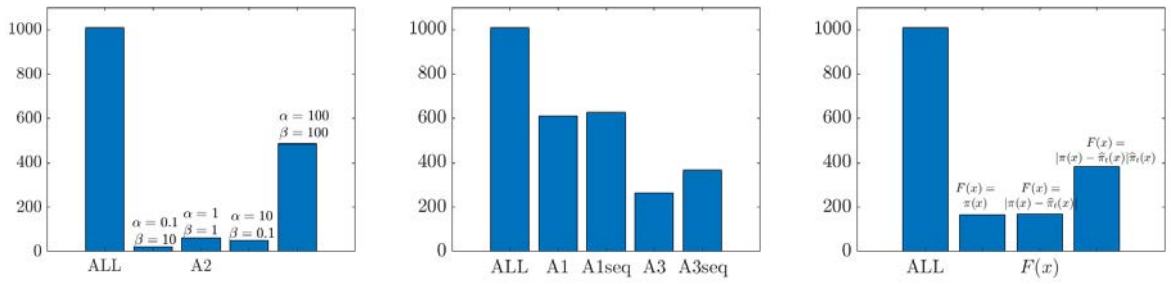


(a) (b) (c)

Figure 5.9: Performance of NN-AIS+U with acceptance function from Eq. (5.26) for different choices of α and β, in **(a)** the estimation of Z. Number of nodes versus t in **(b)** in linear scale, and **(c)** in logarithm scale. The black solid curve represents the number of nodes of NN-AIS+U that accepts all.

formance of the NN-AIS+U accepting all samples. Figure 5.10(a) shows the number of nodes of the final constructed emulators. We see that the choice $\alpha = \beta = 100$ produces an approximation $\widehat{\pi}_t$ that has only half of the nodes of the algorithm accepting all the samples, but achieves the same level of precision in the estimation. We also tested the acceptance functions based on resampling in Eq. (5.28). The results are given in Figures 5.9(c) and 5.10(c). We also tested the acceptance functions **A1** and **A3**, each one with the two possible updating schemes from Tables 5.2 (non-sequential) and 5.3 (sequential). As shown in Figure 5.9(b), the acceptance function **A3** provides better results than **A1**. For both, the use of a sequential updating scheme improve the results. Figure 5.10(b) shows the number of final nodes of the emulator. We can observe that several parsimonious schemes provide very good performance, close to the vanilla NN-AIS+U (with a much smaller number of added nodes).

Finally, in Figure 5.11 we compare the best parsimonious schemes with the vanilla NN-AIS+U method, showing their RMSE as function of the total number of added nodes at each iterations. Furthermore, as the dashed line in Figure 5.6(d), we have compared with an NN-AIS+U scheme where N nodes are added at each iteration but chosen randomly in the space (instead of adding the nodes obtained in the inner resampling in Step 3 of Table 5.1). The corresponding curve is shown with a dashed line. The end point of each curve is highlighted with greater black circle. The reason is that this last point is completely comparable among the different curve since, at this point, we have the same number of target evaluations E. Therefore, observing these last points, we can see that all the parsimonious schemes achieve the same or smaller error than the vanilla NN-AIS+U, with a smaller number of added nodes.



(a) (b) (c)

Figure 5.10: Final number of added nodes for the construction of the emulator for NN-AIS+U with several acceptance functions of different parsimonious schemes.

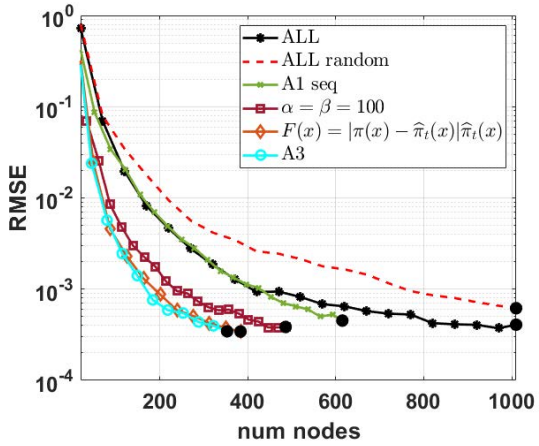


Figure 5.11: RMSE for NN-AIS+U with different acceptance functions (choosing the best schemes in the previous tests), versus the total number of added nodes at each iteration. We have also incorporated a curve (depicted with dashed line) of an NN-AIS+U scheme where N nodes are added at each iteration but chosen randomly in the space (instead of adding the nodes obtained in the inner resampling in Step 3 of Table 5.1). The end point in each curve is highlighted with greater black circle. The reason is that this last point is completely comparable among the different curve since, at this point, we have the same number of target evaluations E. Observing these end points, we see that all the parsimonious schemes shown in the figure provide the same or smaller error than the vanilla NN-AIS+U, with a smaller number of added nodes.

5.8.2. Toy example 2: multimodal density

In this experiment, we consider a multimodal Gaussian target in $d_x = 10$,

$$\bar{\pi}(\mathbf{x}) = \frac{1}{3}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \frac{1}{3}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) + \frac{1}{3}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3),$$

with $\boldsymbol{\mu}_1 = [5, 0, \ldots, 0]$, $\boldsymbol{\mu}_2 = [-7, 0, \ldots, 0]$, $\boldsymbol{\mu}_3 = [1, \ldots, 1]$ and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = 4^2\mathbf{I}_{10}$. We want to test the performance of the different methods in estimating the normalizing constant $Z = 1$. Specifically, our aim is to test the combination of our NN-AIS scheme with an AIS algorithm against other AIS algorithms. The budged of target evaluations is $E = 1000$.

Methods. We consider three sophisticated AIS schemes, namely *population Monte Carlo* (PMC)[10], *layered adaptive IS* (LAIS)[48] and *adaptive multiple IS* (AMIS)[14]. These

are AIS algorithms where the proposal (or proposals) gets updated at each iteration using information from previous samples. Specifically, PMC performs multinomial resampling to locate the proposals in the next iteration; AMIS matches the mean of the single proposal with the current estimation of the posterior mean using all previous samples; LAIS evolves the location parameters of the proposals with a MCMC algorithm. The goal is to compare the performance of PMC, LAIS and AMIS with a combination of our NN-AIS scheme and LAIS.

We set Gaussian pdfs as the proposal pdfs for all methods. We also need to set the number of these proposals in PMC and LAIS, as well as the dispersion of the Gaussian densities. For PMC, we test different number of proposals $N_{\text{PMC}} \in \{10, 100, 200, 500\}$, whose means are initialized at random in $[-15, 15]^{10}$. At each iteration of PMC, one sample is drawn from each of the N_{PMC} proposals, hence the algorithm is run for $T_{\text{PMC}} = \frac{1000}{N_{\text{PMC}}}$ iterations for a fair comparison. As a second alternative, we consider the deterministic mixture weighting approach for PMC, which is shown to have better overall performance, denoted DM-PMC [63, 81].

For LAIS, we also test different number of proposals $N_{\text{LAIS}} \in \{10, 100, 200, 500\}$. We consider the *one-chain* application of LAIS (OC-LAIS), that requires to run one MCMC algorithm targeting $\bar{\pi}(\mathbf{x})$ to obtain the N_{LAIS} location parameters, hence it requires N_{LAIS} evaluations of the target. Then, at each iteration of LAIS, one sample is drawn from the mixture of proposals, hence we run the algorithm for $T_{\text{LAIS}} = 1000 - N_{\text{LAIS}}$ iterations for a fair comparison. For simplicity, we also consider Gaussian random-walk Metropolis to obtain the N_{LAIS} means.

Finally, we consider AMIS with several combinations of number of iterations T_{AMIS} and number of samples per iteration M. At each iteration, M samples are drawn from a single Gaussian proposal, hence the total number of evaluations is $E = MT_{\text{AMIS}}$. In this case, we test $E \in \{1000, 2000, 3000, 5000\}$, so the comparison is not fair (penalizing our approach) except for $E = 1000$.

Regarding our method, we use a mixture of $N_{\text{LAIS}} \in \{100, 200, 500\}$ proposal pdfs obtained by LAIS as $\bar{q}_{\texttt{par}}(\mathbf{x})$ as in Eq. (5.21) (we also use the means of these proposals as initial nodes). We vary N, and run our combined scheme for $T = \frac{E - N_{\text{LAIS}}}{N}$, keeping the number of target evaluations $E = 1000$. For PMC, LAIS and AMIS, as well as for the random walk proposal within the Metropolis algorithm, the covariance of the Gaussian proposals was set to $\xi^2 \mathbf{I}_{10}$ and we test $\xi = 1, ..., 6$. All the methods are compared through the mean absolute error (MAE) in estimating Z, and the results are averaged over 500 independent simulations.

The results are shown in Table 5.5, Table 5.6 and Table 5.7. We can see that NN-AIS+LAIS provides more robust results than only using LAIS. Namely, NN-AIS+LAIS obtains the same or a lower MAE than LAIS, depending on choice of the different parameters. Overall, the proposed scheme outperforms all the other benchmark AIS methods such as PMC, DM-PMC, LAIS and AMIS easily, even considering more target evaluations (penalizing our scheme) as shown in Table 5.7.

Table 5.5: **MAE for Z with $E = 1000$** (best and worst MAE of each method are boldfaced)

Methods		$\xi = 1$	$\xi = 2$	$\xi = 3$	$\xi = 4$	$\xi = 5$	$\xi = 6$
PMC	$N_{\text{PMC}} = 10$	0.9993	0.9526	0.8603	0.6743	0.6024	0.6155
	$N_{\text{PMC}} = 100$	0.9998	0.9896	0.8853	0.6761	0.5192	**0.4544**
	$N_{\text{PMC}} = 200$	**1.0002**	0.9893	0.8816	0.7099	0.6389	0.5384
	$N_{\text{PMC}} = 500$	0.9995	0.9916	0.9741	0.8700	0.7421	0.6544
DM-PMC	$N_{\text{PMC}} = 10$	0.9991	0.9478	0.8505	0.6009	0.5352	0.5814
	$N_{\text{PMC}} = 100$	0.9997	0.8719	0.4490	0.2425	**0.1901**	0.2193
	$N_{\text{PMC}} = 200$	0.9999	0.9321	0.5708	0.3257	0.2374	0.2524
	$N_{\text{PMC}} = 500$	**1.0000**	0.9888	0.7969	0.5009	0.3684	0.3800
OC-LAIS	$N_{\text{LAIS}} = 10$	**1.0000**	1.0000	0.9992	0.9883	0.9468	0.9079
	$N_{\text{LAIS}} = 100$	0.9999	0.8731	0.4434	0.2785	0.2392	0.2870
	$N_{\text{LAIS}} = 200$	0.9982	0.7028	0.2418	0.1243	0.1406	0.2070
	$N_{\text{LAIS}} = 500$	0.9937	0.4949	0.1221	**0.0857**	0.1195	0.1786

Table 5.6: **MAE for Z with $E = 1000$** (best of each combination of N_{LAIS} and ξ are boldfaced)

Methods		$\xi = 1$	$\xi = 2$	$\xi = 3$	$\xi = 4$	$\xi = 5$	$\xi = 6$
NN-AIS+LAIS ($N_{\text{LAIS}} = 100$)	$N = 50$	**0.9778**	**0.3886**	**0.1334**	0.1487	**0.1624**	**0.1968**
	$N = 100$	0.9900	0.4152	0.1408	0.1519	0.1853	0.2502
	$N = 300$	0.9907	0.4817	0.1761	**0.1466**	0.1869	0.2427
NN-AIS+LAIS ($N_{\text{LAIS}} = 200$)	$N = 100$	**0.7662**	**0.1607**	0.1332	**0.1179**	**0.1300**	0.2000
	$N = 200$	0.8195	0.2176	**0.1001**	0.1250	0.1418	**0.1854**
	$N = 400$	0.8417	0.2954	0.1512	0.1218	0.1522	0.2060
NN-AIS+LAIS ($N_{\text{LAIS}} = 500$)	$N = 50$	**0.2428**	0.1801	0.1614	0.1313	0.1190	0.1642
	$N = 100$	0.2905	0.1406	**0.1144**	0.1046	**0.1152**	0.1851
	$N = 250$	0.4139	**0.1270**	0.1226	**0.0989**	0.1262	**0.1783**

5.8.3. Inference in an Astronomical model

In recent years, the problem of revealing objects orbiting other stars has acquired large attention in Astronomy. Different techniques have been proposed to discover exo-objects but, nowadays, the radial velocity technique is still the most used [30, 3, 80]. The model is highly non-linear and it is costly in terms of computation time (specially, for certain sets of parameters). The evaluation of the posterior involves numerically integrating a differential equation in time or an iterative procedure for solving a non-linear equation. Typically, the iteration is performed until a threshold is reached, or a certain number of iterations (e.g., typically 10^6 iterations), are performed. For the radial velocity model, this

Table 5.7: **MAE for Z of AMIS with $E \in \{1000, 2000, 3000, 5000\}$.** Note that the comparison is unfair (penalizing our approach) except for $E = 1000$.

Methods		$\xi = 1$	$\xi = 2$	$\xi = 3$	$\xi = 4$	$\xi = 5$	$\xi = 6$
	$M = 10$	0.9998	0.9997	0.9997	0.9996	0.9996	0.9995
AMIS	$M = 100$	1.0000	1.0000	1.0000	0.9999	0.9997	0.9990
$E = 1000$	$M = 200$	1.0000	1.0000	1.0000	1.0000	0.9998	0.9994
	$M = 500$	**1.0000**	1.0000	1.0000	1.0000	0.9998	**0.9989**
	$M = 10$	0.9155	0.9117	0.8981	0.8987	0.8891	**0.8878**
AMIS	$M = 100$	0.9998	0.9986	0.9934	0.9784	0.9559	0.9072
$E = 2000$	$M = 200$	1.0000	1.0000	0.9998	0.9981	0.9888	0.9712
	$M = 500$	**1.0000**	1.0000	1.0000	0.9998	0.9984	0.9953
	$M = 10$	0.3293	0.3402	**0.3051**	0.3381	0.3540	0.3443
AMIS	$M = 100$	0.9725	0.9040	0.7963	0.6384	0.4964	0.3816
$E = 3000$	$M = 200$	0.9998	0.9977	0.9884	0.9527	0.8308	0.7119
	$M = 500$	**1.0000**	1.0000	0.9998	0.9988	0.9859	0.9566
	$M = 10$	0.0766	0.0768	**0.0695**	0.0722	0.0699	0.0725
AMIS	$M = 100$	0.1626	0.1176	0.0957	0.0810	0.0737	0.0656
$E = 5000$	$M = 200$	0.8771	0.6040	0.2824	0.1473	0.1163	0.0899
	$M = 500$	**1.0000**	0.9982	0.9904	0.9449	0.7944	0.4532

is needed for solving Eq. (5.36) described below. In the following, we describe an orbital model, which is equivalent for any N-body system observed from Earth, i.e. exoplanetary systems, binary stellar system, double pulsars, etc.

Likelihood function and prior densities

When analysing radial velocity data of an exoplanetary system, it is commonly accepted that the *wobbling* of the star around the centre of mass is caused by the sum of the gravitational force of each planet independently and that they do not interact with each other. Each planet follows a Keplerian orbit and the radial velocity of the host star is given by

$$y_k = V_0 + \sum_{i=1}^{S} \zeta_i \left[\cos\left(u_{i,k} + \omega_i\right) + e_i \cos\left(\omega_i\right)\right] + \xi_k, \tag{5.33}$$

with $k = 1, \ldots, K$. The number of objects in the system is S, that is consider known in this experiment (for the sake of simplicity). Note that the iteration index $i = 1, ..., S$ denotes the i-th object/planet. Both y_k, $u_{i,k}$ depend on time t, and ξ_k is a Gaussian noise perturbation with variance σ_e^2. For simplicity, we consider this value known, $\sigma_e^2 = 1$. The meaning of each parameter in Eq. (5.33) is given in Table 5.8. The likelihood function is defined by (5.33) and some indicator variables described below. The angle $u_{i,k}$ is the true

Table 5.8: Description of parameters in Eq. (5.33).

Parameter	Description	Units
For each planet		
ζ_i	amplitude of the curve	$\mathrm{m\,s^{-1}}$
$u_{i,k}$	true anomaly	rad
ω_i	longitude of periastron	rad
e_i	orbit's eccentricity	...
P_i	orbital period	s
τ_i	time of periastron passage	s
Below: not depending on the number of objects/satellite		
V_0	mean radial velocity	$\mathrm{m\,s^{-1}}$

anomaly of the planet i and it can be determined from

$$\frac{du_{i,k}}{dt} = \frac{2\pi}{P_i}\frac{(1+e_i\cos u_{i,k})^2}{(1-e_i)^{\frac{3}{2}}} \tag{5.34}$$

This equation has analytical solution. As a result, the true anomaly $u_{i,k}$ can be determined from the mean anomaly $M_{i,k}$. However, the analytical solution contains a non linear term that needs to be determined by iterating. First, we define the mean anomaly $M_{i,k}$ as

$$M_{i,k} = \frac{2\pi}{P_i}(t-\tau_i), \tag{5.35}$$

where τ_i is the time of periastron passage of the planet i and P_i is the period of the orbit (see Table 5.8). Then, through the Kepler's equation,

$$M_{i,k} = E_{i,k} - e_i \sin E_{i,k}, \tag{5.36}$$

we have to obtain $E_{i,k}$, which is the eccentric anomaly. Equation (5.36) has no analytic solution and it must be solved by an iterative procedure. A Newton-Raphson method is typically used to find the roots of this equation [66]. For certain sets of parameters this iterative procedure can be particularly slow.

Finally, we can also obtain $u_{i,k}$ from

$$\tan\frac{u_{i,k}}{2} = \sqrt{\frac{1+e_i}{1-e_i}}\tan\frac{E_{i,k}}{2}, \tag{5.37}$$

Hence, the vector of variables to infer, $\mathbf{x}$, is

$$\mathbf{x} = [V_0, \zeta_1, \omega_{1,t}, e_1, P_1, \tau_1, \ldots, \zeta_S, \omega_S, e_S, P_S, \tau_S], \tag{5.38}$$

For a single object (e.g., a planet or a natural satellite), the dimension of $\mathbf{x}$ is $d_X = 5{+}1 = 6$, with two objects the dimension of $\mathbf{x}$ is is $d_x = 11$ etc. Generally, we have $d_x = 1 + 5S$. Note that the observation model in Eq. (5.33) induces the likelihood function $p(\mathbf{y}|\mathbf{x})$,

where $\mathbf{y} = [y_1, ..., y_K]$.

Priors. As prior densities we consider uniform pdfs in the following intervals: $V_0 \in [-20, 20]$, $\zeta_i \in [0, 50]$, $e_i \in [0, 1]$, $P_i \in [0, 365]$, $\omega_{i,k} \in [0, 2\pi]$, $\tau_i \in [0, P_i]$ (i.e., the prior is zero outside these intervals), for all $i = 1, \ldots, S$. This means that the likelihood function is zero when the particles fall out of these intervals. Note that the interval of τ_i is conditioned to the value P_i. This parameter is the time of periastron passage, i.e. the time passed since the object passed the closest point in its orbit. It has the same units of P_i and can take values from 0 to P_i.

Experiment setting and results

We generate a set of data $\{y_k\}_{t=1}^K$ with $K = 50$, and $S = 2$ objects (so that $d_x = 11$), according to the observation model above. We set $V = 2$, $\zeta_1 = 25$, $\omega_1 = 0.61$, $e_1 = 0.1$, $P_1 = 15$, $\tau_1 = 3$ (for the first object) and $\zeta_2 = 5$, $\omega_2 = 0.17$, $e_2 = 0.3$, $P_2 = 115$, $\tau_2 = 25$ (for the second object). We compare a standard IS scheme using the prior as proposal and the NN-AIS+U scheme (using again the prior as uniform proposal component) using the parsimonious scheme with acceptance function A3 in Eq. (5.27). In NN-AIS+U, we consider $N = 10000$, $T = 100$ and $L = 10^6$. The total number of evaluations of the posterior is then $NT = 10^6$ for NN-AIS+U. For the standard IS scheme, we consider different number of samples $\{10^6, 2 \cdot 10^6, 3 \cdot 10^6, 4 \cdot 10^6\}$. We compute the Relative MSE (RMSE) in estimation of the 11 parameters in $\mathbf{x}$, averaged over all the components. The results are also averaged over 200 independent runs. Table 5.9 provides the RMSE and the computational time, normalized with respect to the time spent by the standard IS scheme with 10^6 samples. We can observe that, in order to obtain the same performance of NN-AIS+U in terms of RMSE, the IS schemes require much more computational time than NN-AIS+U. Therefore, this is an example with a real-world model where the inequality (5.22) is fulfilled.

Table 5.9: Relative Mean Square Errors (MSE) and normalized computational time.

Methods	**NN-AIS+U**	**IS**	**IS**	**IS**	**IS**
RMSE	5.755	9.439	7.943	6.524	5.431
normalized time	1.53	1	1.91	3.20	4.17
posterior evaluations (E)	10^6	10^6	$2 \cdot 10^6$	$3 \cdot 10^6$	$4 \cdot 10^6$

5.8.4. Retrieval of biophysical parameters inverting an RTM model

In this experiment, we apply NN-AIS to retrieve biophysical parameters of a sequence of problems involving the radiatrive transfer PROSAIL model. The purpose is to show the ability of NN-AIS to share information from related inverse problems easily. The

combined PROSPECT leaf optical properties model and SAIL canopy bidirectional reflectance model, also referred to as PROSAIL, have been used for almost two decades to study plant canopy spectral and directional reflectance in the solar domain [33]. PROSAIL has also been used to develop new methods for retrieval of vegetation biophysical properties. It links the spectral variation of canopy reflectance, which is mainly related to leaf biochemical contents, with its directional variation, which is primarily related to canopy architecture and soil/vegetation contrast. This link is key to simultaneous estimation of canopy biophysical/structural variables for applications in agriculture, plant physiology, and ecology at different scales. PROSAIL has become one of the most popular radiative transfer tools due to its ease of use, general robustness, and consistent validation by lab/field/space experiments over the years.

Inversion of PROSAIL. The context is Bayesian inversion of an observation model $\mathbf{h}(\mathbf{x})$.[8] In our setting, the observation model is PROSAIL, which models reflectance in terms of leaf optical properties and canopy level characteristics. We choose only leaf optical properties as the set parameters of interest

$$\mathbf{x} = [S_{st}, C_{hl}, C_{ar}, C_{br}, C_w, C_m] \in \mathbb{R}^6, \tag{5.39}$$

described in Table 5.10. In Table 5.11, we show the fixed values of canopy level characteristics, which are determined by the leaf area index (LAI), the average leaf angle inclination (ALA), the hot-spot parameter (Hotspot), and the parameters of system geometry described by the solar zenith angle (θ_s), view zenith angle (θ_v), and the relative azimuth angle between both angles ($\Delta\Theta$). The observation model is $\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v}$, where $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_{d_y})$ with $\sigma = 1$. The observed data, denoted $\mathbf{y} \in \mathbb{R}^{d_y}$ with $d_y = 2101$, corresponds to the detected spectra. We generated synthetic spectra and the goal is to infer $\mathbf{x}$ studying the corresponding posterior distribution. The Gaussian noise $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ jointly with PROSAIL, $\mathbf{h}(\mathbf{x})$, induces the following likelihood function

$$\ell(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{h}(\mathbf{x}), \sigma^2\mathbf{I}). \tag{5.40}$$

We set the prior $g(\mathbf{x})$ as a product of indicator variables $S_{st} \in [1, 3]$, $C_{hl} \in [0, 100]$, $C_{ar} \in [0, 25]$, $C_{br} \in [0, 1]$, $C_w \in [0, 0.05]$ and $C_b \in [0, 0.02]$, i.e., the prior is zero outside these intervals.[9] The complete posterior is then $p(\mathbf{x}|\mathbf{y}) = \frac{1}{Z}\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})$. It is important to remark that PROSAIL is an *highly non-linear* model and its inversion is a very complicated problem, as shown in the remote sensing literature [8, 7].

Sequential inversion for image recovery. In remote sensing, the goal is usually to recover an image formed by R pixels. A set of physical parameters $\mathbf{x}_r$ is associated to the r-th pixel. Hence, the corresponding vector of observations $\mathbf{y}_r$ is also associate to each pixel. We have then a collection of inverse problems, where we desire to retrieve $\mathbf{x}_r$

[8] The MATLAB code of PROSAIL is available in http://teledetection.ipgp.jussieu.fr/prosail/.

[9] We have employed the ranges suggested http://opticleaf.ipgp.fr/index.php?page=prospect.

given $\mathbf{y}_r$, one for each pixel. Mathematically, let consider R measurements, $\{\mathbf{y}_r\}_{r=1}^R$, associated each to a different inverse problem, under the PROSAIL model, i.e., a mapping $\mathbf{h}(\mathbf{x}) : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$,

$$\mathbf{y}_r = \mathbf{h}(\mathbf{x}_r) + \mathbf{v}_r, \quad r = 1, \ldots, R. \tag{5.41}$$

We assume $\mathbf{v}_r \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{d_y})$, for all $r = 1, \ldots, R$, with $d_y = 2101$, and $\sigma = 1$, and thus we have a R posterior distributions $p_r(\mathbf{x}_r|\mathbf{y}_r)$ for $r = 1, \ldots, R$ (we recall that $\mathbf{x}_r \in \mathbb{R}^{d_x}$,with $d_x = 6$). We solve them sequentially while reusing information. Some examples of data $\mathbf{y}_r$ and model values are given in Figure 5.12.

Table 5.10: Description of parameters in Eq. (5.39).

Parameter	Description	Units
S_{st}	structure coefficient	—
C_{hl}	chlorophyll content	μg cm^{-2}
C_{ar}	carotenoid content	μg cm^{-2}
C_{br}	brown pigment content	—
C_w	water content	cm
C_m	dry matter content	g cm^{-2}

Table 5.11: Characteristics of the simulation used in the PROSAIL model.

Canopy level	LAI	ALA	Hotspot	θ_s	θ_v	$\Delta\Theta$
	5	30	0.01	30	10	90

Experiment. In real data settings, physical and geographical patterns are associated to the parameters $\mathbf{x}_r$ in the image. In order to check the performance of each algorithm, we consider synthetic data. Thus, in this experiment, we have also generated synthetic patterns in order to simulate a real scenario. In particular, we produce six patterns (recall $\mathbf{x} \in \mathbb{R}^6$) that represent handwritten digits (see Figure 5.13). Hence, in this setting, we have $R = 784$ different observation vectors $\mathbf{y}_r$, $r = 1, \ldots, R$, for which we want to estimate the vectors of true values $\mathbf{x}_r$, $r = 1, \ldots, R$. Each observation corresponds to a single pixel of a 28×28 image. We also compute the maximum a-posteriori (MAP) of $p_r(\mathbf{x}_r|\mathbf{y}_r)$, $\mathbf{x}_{r,\text{MAP}}$, as estimate of $\mathbf{x}_r$.

Methods. We use the NN-AIS scheme to estimate $\mathbf{x}_{r,\text{MAP}}$ for $r = 1, \ldots, 784$, and compare it against IS using the prior as proposal density, in terms of relative squared error and by looking at the recovered images. As parameters of our scheme we chose $N_{\text{init}} = 1000$, $T = 20$, $N = 250$ and $L = 10^5$. The N_{init} initial points were taken at random in the domain except for 11 points that were placed in the vertices of the domain. Our scheme allows for sharing information from problem to the next one, so we also use the $\widehat{\mathbf{x}}_{s,\text{MAP}}$ for $s = 1, \ldots, r-1$ as initial nodes when estimating $\mathbf{x}_{r,\text{MAP}}$. Note that this is completely fair since the model has been already evaluated at those points. The comparison is fair in terms of model evaluations, with a total of $E = 6000$ for each $r = 1, \ldots, 784$.

Results. The results are shown in Figures 5.14 and 5.15. It can be seen that both standard IS and NN-AIS are able to correctly recover components 2, 4, 5 and 6 of $\mathbf{x}_r$ ($r = 1, \ldots, 784$), i.e., the images of "2", "4", "5" and "6" in both Figure 5.14 and Figure 5.15 look very close to the true ones (Figures 5.13(b),(d),(e) and (f) respectively). The images recovered by NN-AIS have lower noise though. The components 1 and 3 of the $\mathbf{x}_r$'s are completely lost with standard IS (see Figure 5.14), whereas NN-AIS is able at least to achieve to recover the boundaries of the corresponding patterns. Indeed, NN-AIS obtains a much lower error in estimation, as it is shown in Table 5.12 and Table 5.13. The difficulty in recovering the components 1 (i.e., S_{st}) and 3 (i.e., C_{ar}) deserves further studies. This issue could be related to some relevant features of PROSAIL (e.g., the average partial derivatives with respect to these two components). We leave the study of these specific issues for future work. In Table 5.14, we also show the averaged error in the spectra produced by both methods as compared to the true observations.

Table 5.12: Relative Mean Absolute Errors (RMAE) for each component (averaged over all spectra).

Components	1	2	3	4	5	6	**Mean**
Stand. IS	0.7556	0.4397	2.9431	0.6247	0.2096	0.2782	**2.8516**
Sequential NN-AIS	0.2045	0.2245	0.8891	0.1985	0.1425	0.1320	**1.0715**

Table 5.13: Mean Absolute Errors (RMAE) for each component (averaged over all spectra).

Components	1	2	3	4	5	6	**Mean**
Stand. IS	0.9760	6.1754	9.8204	0.1348	0.0016	0.0012	**0.8752**
Sequential NN-AIS	0.2641	3.1535	2.9667	0.0428	0.0011	0.0006	**0.2985**

Table 5.14: Absolute and relative error (averaged over all the pixels) in the transformed domain ("reconstruction of the spectra")

	Absolute	**Relative**
Stand. IS	66.4395	0.0802
Sequential NN-AIS	11.0844	0.0198

5.9. Conclusions and future lines

In this work, we introduced a novel framework of adaptive importance sampling algorithms. The key idea is the use of a non-parametric proposal density built by a regression procedure (the emulator), that mimics the true shape of posterior pdf. Hence, the proposal pdf represents also a surrogate model, that is in turn adapted through the iterations

by adding new support points. The regression (e.g., obtained by nearest neighbors and Gaussian processes) can be applied directly on the posterior domain or, alternatively, in just one piece of the likelihood, such as an arbitrary physical model. Drawing from the emulator is possible by a deep architecture of two nested IS layers. More sophisticated deep structures, employing a a chain of emulators, have been described.

RADIS is an extremely efficient importance sampling scheme since the emulator (used as proposal pdf) becomes closer and closer to the true posterior, as new nodes are incorporated. As a consequence, RADIS asymptotically converges to an exact sampler under mild conditions. Several numerical experiments and theoretical supports confirm these statements. Robust accelerating versions of RADIS have been also presented, as well as combinations with other benchmark AIS algorithms. Cheap constructions of the emulator have been also discussed and tested. The use of RADIS within a sequential Monte Carlo scheme will be considered in future works. Furthermore, as future research lines, we also plan to analyze in depth the PROSAIL inversion problem, approximating the partial derivatives with respect some specific parameters by RADIS. Moreover, we also plan to consider the adaptation of the auxiliary proposal $\bar{q}_{\text{aux}}(\mathbf{x})$, adding also additional layers in the proposed deep architecture.

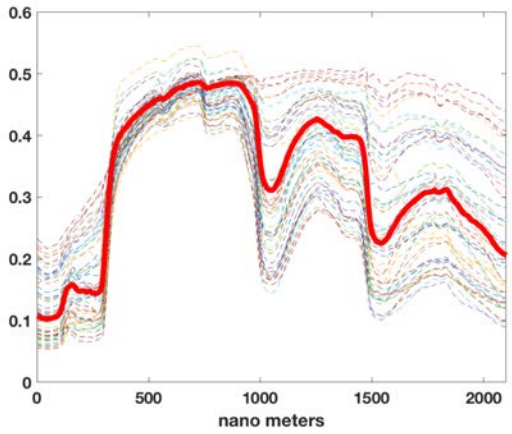
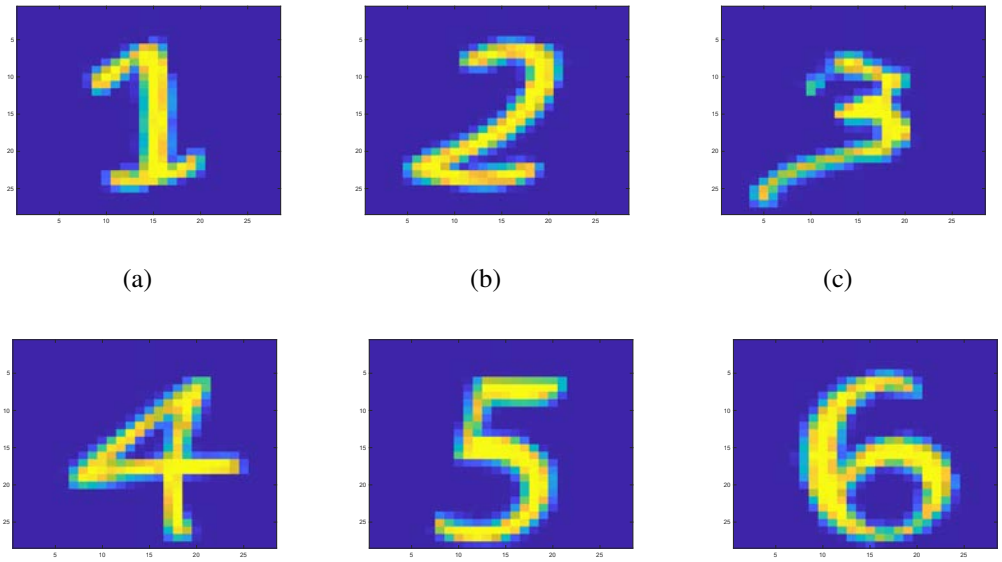


Figure 5.12: An example of vector of data $\mathbf{y}_r$ (hyperspectral reflectances, shown with solid line) and the model values corresponding to 50 different samples, $\mathbf{f}^{(i)} = \mathbf{f}(\mathbf{x}_r^{(i)})$ (dashed lines). Each component of the vector $\mathbf{y}_r$ corresponds to a different wavelength (nm).

Acknowledgements

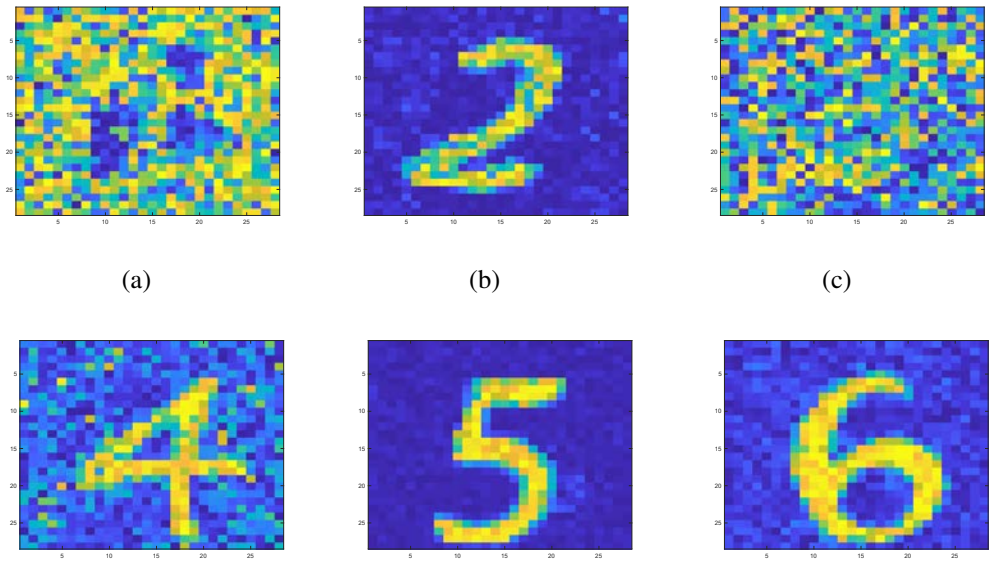
This work has been supported by Spanish government via grant FPU19/00815.



(a) (b) (c)

(d) (e) (f)

Figure 5.13: Patterns of the true parameter values (scaled according to range of each parameter), i.e., the ground-truths.



(a) (b) (c)

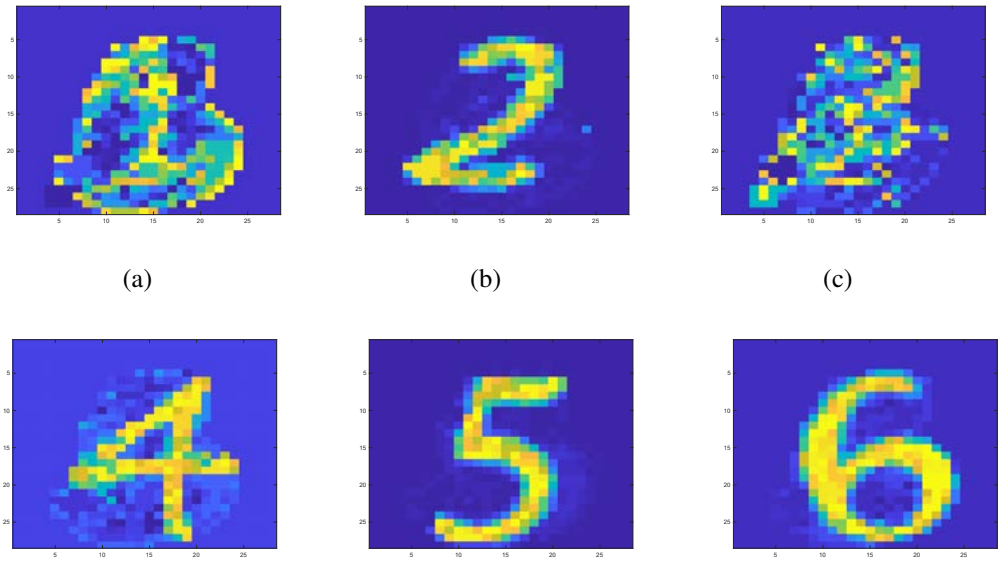
(d) (e) (f)

Figure 5.14: Recovered by standard IS. We can observe the difficulty in the retrieval of the first and third parameter.

5.10. Appendix

5.10.1. Theoretical support

In this section, we discuss several theoretical aspects of RADIS. First, we address the error in the approximate sampling and evaluation of the interpolating proposal. Then, we



(a) (b) (c)

(d) (e) (f)

Figure 5.15: Recovered by NN-AIS. We can observe the retrieval of first and third parameter is not completely successful.

show that the adaptive construction of the proposal decreases the distance with respect to the true target as the number of nodes J_t grows. Finally, we show that this also minimizes the variance of the IS weights.

Sampling Importance Resampling (SIR)

Let $\widehat{\pi}_t(\mathbf{x})$ the unnormalized interpolating proposal from which we aim to sample. Its normalizing constant c_t is not important in this first part. The SIR method allows to sample from the density $\widehat{\pi}_t$ by resampling a sample drawn from another auxiliary (importance) density [73][25, Chapter 24]. This method is also referred as the weighted bootstrap in [75, Sect. 3.2]. The SIR algorithm is as follows:

1. Draw $\{\mathbf{x}_1, \ldots, \mathbf{x}_L\}$ i.i.d. from $\bar{q}_{\text{aux}}(\mathbf{x})$, that is a density with fatter tails than $\widehat{\pi}_t(\mathbf{x})$.
2. Calculate the importance weights for each $\mathbf{x}_i$

$$\gamma_i = \gamma(\mathbf{x}_i) = \frac{\widehat{\pi}_t(\mathbf{x}_i)}{\bar{q}_{\text{aux}}\mathbf{x}_i)}.$$

3. Resample N ($N \leq L$) values $\{\mathbf{x}_1^*, \ldots, \mathbf{x}_N^*\}$ from $\{\mathbf{x}_1, \ldots, \mathbf{x}_L\}$ with probabilities proportional to γ_i assigned to $\mathbf{x}_i$.

If $L \to \infty$, or more precisely $\frac{L}{N} \to \infty$, then the set $\{\mathbf{x}_1^*, \ldots, \mathbf{x}_N^*\}$ is asymptotically distributed as $\widehat{\pi}_t(\mathbf{x})$. Thus, the choice of L and N is important for two factors: (i) to reduce the dependence of the $\mathbf{x}_i^*$'s, and (ii) to have the distribution of $\mathbf{x}_i^*$ as close to $\widehat{\pi}_t$ as possible.

The relative magnitude between N and L controls this dependence, while only the magnitude of L affects how close the distribution of the resampled particle is to the density $\widehat{\pi}_t$.

Bias and correlation in SIR. Under mild conditions, as $\frac{L}{N} \to \infty$, the density of resampled particle converges to $\widehat{\pi}_t(\mathbf{x})$. For more details see [28, Sect. 6.2.4], [75, Sect. 3.2] and [72, page 6]. As SIR is an approximate sampling algorithm, it has some bias[10]. If the first and second moment of the IS weight $\gamma(\mathbf{x}) = \frac{\widehat{\pi}(\mathbf{x}_i)}{\bar{q}_{\text{aux}}(\mathbf{x})}$ exists, it can be shown that this bias vanishes at $O(L^{-1})$ rate [25, Chapter 24]. In [75, Sect. 3.2], they show the convergence of the cdf of the resampled particle as $L \to \infty$ in the the univariate case.
Resampling N times from a unique pool of L samples from $\bar{q}_{\text{aux}}(\mathbf{x})$ introduces correlation in the resampled sample. However, when $N \ll L$, this correlation is negligible. Some heuristics suggest $\frac{L}{N} = 20$ [73], or $\frac{L}{N} \geq 10$ [75]. For more details in the relation of the values of L and N see [25][Sect. 24.3]. In [45] (see Figure 5 and Appendix A therein), it is shown the "equivalent" density of a resampled particle for a fixed value of L, which converges to the target pdf as L diverges. Furthermore, for computing the denominator of the outer weights we need the normalizing constant of $\widehat{\pi}_t$ (see Eq. (5.11)). In this sense, the inner IS also provides with an approximation by using the L samples from $\bar{q}_{\text{aux}}(\mathbf{x})$,

$$\widehat{c}_t = \frac{1}{L} \sum_{\ell=1}^{L} \frac{\widehat{\pi}_t(\mathbf{z}_\ell)}{\bar{q}_{\text{aux}}(\mathbf{z}_\ell)}. \tag{5.42}$$

This estimate converges as $L \to \infty$ [71].

Variance of the IS weights

Let $w(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\bar{q}(\mathbf{x})}$ be the weight function evaluated at samples $\mathbf{x} \sim \bar{q}(\mathbf{x})$. First of all, note that $E[w(\mathbf{x})] = Z$. Below, we show that the variance of $w(\mathbf{x})$ is proportional to the Pearson divergence between the posterior $\bar{\pi}$ and proposal q, i.e.,

$$\begin{aligned}
\texttt{var}[w(\mathbf{x})] &= \int_{\mathcal{X}} (w(\mathbf{x}) - Z)^2 \bar{q}(\mathbf{x}) d\mathbf{x} && (5.43)\\
&= \int_{\mathcal{X}} \left(\frac{\pi(\mathbf{x}) - Z\bar{q}(\mathbf{x})}{\bar{q}(\mathbf{x})} \right)^2 \bar{q}(\mathbf{x}) d\mathbf{x} && (5.44)\\
&= Z^2 \int_{\mathcal{X}} \frac{(\bar{\pi}(\mathbf{x}) - \bar{q}(\mathbf{x}))^2}{\bar{q}(\mathbf{x})} d\mathbf{x} = Z^2 \chi^2(\bar{\pi} \| q), && (5.45)
\end{aligned}$$

where $\chi^2(\bar{\pi}\|q) = \int_{\mathcal{X}} \frac{(\bar{\pi}(\mathbf{x}) - \bar{q}(\mathbf{x}))^2}{\bar{q}(\mathbf{x})} d\mathbf{x}$, is the Pearson divergence and we have used $\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$. Hence, if we construct a proposal such $\chi^2(\bar{\pi}\|q) \to 0$, we would obtain $\texttt{var}[\widehat{Z}] = 0$. Moreover, the mean square error (MSE) of $\widehat{I}$ can also be shown to be bounded by this divergence (see e.g. [1])

$$\mathbb{E}[|I - \widehat{I}|^2] \leq \frac{C_f(\chi^2(\bar{\pi}\|\bar{q}) + 1)}{N}. \tag{5.46}$$

Thus, it is beneficial to reduce the $\chi^2(\bar{\pi}\|\bar{q})$ in order to obtain accurate IS estimators.

[10] Measured as the difference in the probability of some set between the target pdf and the "equivalent" pdf

Pearson divergence and L_p distances

Now, we aim to show that $\chi^2(\bar{\pi}||\bar{q})$ can be bounded in terms of the L_2 and L_∞ distances, between $\bar{\pi}(\mathbf{x})$ and $\bar{q}(\mathbf{x})$. Using Holder's inequality and the fact that pdfs are always positive, we can write

$$\chi^2(\bar{\pi}||\bar{q}) = \int_{\mathcal{X}} |\bar{\pi}(\mathbf{x}) - \bar{q}(\mathbf{x})| \frac{|\bar{\pi}(\mathbf{x}) - \bar{q}(\mathbf{x})|}{|\bar{q}(\mathbf{x})|} d\mathbf{x} = \left\| (\bar{\pi} - q) \left(\frac{\bar{\pi} - \bar{q}}{\bar{q}} \right) \right\|_{L_1}$$
$$\leq \|\bar{\pi} - \bar{q}\|_{L_2} \left\| \frac{\bar{\pi} - \bar{q}}{\bar{q}} \right\|_{L_2}. \tag{5.47}$$

The L_2 distance can be easily shown to be bounded by L_∞ distance (considering a bounded domain $\mathcal{X}$), i.e.,

$$\|\bar{\pi} - \bar{q}\|_{L_2} = \left(\int_{\mathcal{X}} |\bar{\pi}(\mathbf{x}) - \bar{q}(\mathbf{x})|^2 d\mathbf{x} \right)^{\frac{1}{2}} \leq \left(|\mathcal{X}| \max |\bar{\pi}(\mathbf{x}) - \bar{q}(\mathbf{x})|^2 \right)^{\frac{1}{2}}$$
$$= |\mathcal{X}|^{\frac{1}{2}} \|\bar{\pi} - \bar{q}\|_{L_\infty}. \tag{5.48}$$

Similarly, we have

$$\left\| \frac{\bar{\pi} - \bar{q}}{\bar{q}} \right\|_{L_2} \leq |\mathcal{X}|^{\frac{1}{2}} \left\| \frac{\bar{\pi} - \bar{q}}{\bar{q}} \right\|_{L_\infty}. \tag{5.49}$$

Thus, we can obtain the following result regarding the L_∞ distance,

$$\chi^2(\bar{\pi}||\bar{q}) \leq |\mathcal{X}| \left\| \frac{\bar{\pi} - \bar{q}}{\bar{q}} \right\|_{L_\infty} \|\bar{\pi} - \bar{q}\|_{L_\infty}. \tag{5.50}$$

Since we choose $\bar{q}$ in order to have fatter tails than $\bar{\pi}$ and since $\bar{q}$, $\bar{\pi}$ are bounded, then the factor $\|\frac{\bar{\pi}-\bar{q}}{\bar{q}}\|_{L_\infty}$ in (5.50) vanishes to zero if $\|\bar{\pi} - \bar{q}\|_{L_\infty} \to 0$. Therefore, if $\|\bar{\pi} - \bar{q}\|_{L_\infty} \to 0$, we have $\chi^2(\bar{\pi}||\bar{q}) \to 0$. Due to (5.48)-(5.49), this result is also valid for the L_2 distance. In this work, we consider $\bar{q} = \bar{q}_t = \frac{1}{c_t}\widehat{\pi}_t$ such $\|\pi - \widehat{\pi}_t\|_{L_\infty} \to 0$ as $t \to \infty$ (see section below), and thus $\|\bar{\pi} - \frac{1}{c_t}\widehat{\pi}_t\|_{L_\infty} \to 0$, that implies $\chi^2(\bar{\pi}||\frac{1}{c_t}\widehat{\pi}_t) \to 0$.

Convergence of the emulator to target function

For simplicity, let us focus on the interpolation case and a bounded $\mathcal{X}$. Here, we show that the interpolating constructions of Sect. 5.4.2, jointly with the adaptation process, lead to a proposal $\frac{1}{\widehat{c}_t}\widehat{\pi}_t(\mathbf{x})$ that converges to $\bar{\pi}(\mathbf{x})$. Since $\widehat{c}_t$ is an unbiased estimation of the area below $\widehat{\pi}_t(\mathbf{x})$, we focus on the convergence of $\widehat{\pi}_t(\mathbf{x})$ to the unnormalized posterior $\pi(\mathbf{x})$. As $\widehat{\pi}_t(\mathbf{x}) \to \pi(\mathbf{x})$, then $\widehat{c}_t \to Z$. In Sect. 5.5, we have introduced an extra parametric density $\bar{q}_{\texttt{par}}(\mathbf{x})$ to also ensure that new points can be added in any region of the domain $\mathcal{X}$ during the adaptation. We show below that, when using the NN or GP constructions, the approximation error of $\widehat{\pi}_t$ depends on a quantity called *fill distance*,

$$r_t = \max_x \min_{i=1,\ldots,J_t} \|\mathbf{x} - \mathbf{x}_i\|_2, \tag{5.51}$$

which measures the filling of the space. In other words, the greater the fill distance, the less covered the space is. For both constructions, decreasing the fill distance ensures that $\widehat{\pi}_t(\mathbf{x})$ converges in L_∞ norm to $\pi(\mathbf{x})$. Using a $\bar{q}_{\texttt{par}}(\mathbf{x})$ that is not negative in $\mathcal{X}$ ensures every region will be covered eventually, i.e., $r_t \to 0$ as $t \to \infty$.

NN construction. If π is Lipschitz continuous, we have that

$$\left\|\pi - \widehat{\pi}_t\right\|_\infty \leq L_0 r_t, \tag{5.52}$$

where L_0 is the Lipschitz constant and r_t denotes the fill distance [40][App. D.4]. Equivalently, we have [6]

$$\left\|\pi - \widehat{\pi}_t\right\|_\infty \leq L_0 \max_{i=1,\dots,J_t} \text{diam}(\mathcal{R}_i), \tag{5.53}$$

that is, the approximation error is bounded by the biggest Voronoi cell. Covering the space (not necessarily with uniform points) ensure that $\max_i \text{diam}(\mathcal{R}_i) \to 0$ [15] (equivalently $r_t \to 0$), and thus $\widehat{\pi}_t \to \pi$ as $t \to \infty$.

GP construction. First, we recall a result valid when the GP regression is applied on π, not a transformation. It can be shown that the approximation error $\|\pi - \widehat{\pi}_t\|_\infty$ is bounded in terms of the fill distance (e.g. see [40][Sect. 7] and references therein)

$$\|\pi - \widehat{\pi}_t\|_\infty = \mathcal{O}(\lambda(r_t)). \tag{5.54}$$

The speed of convergence, i.e., the functional form of $\lambda(r_t)$, depends on the choice of kernel (e.g. under some circumstances and with Gaussian kernel, $\lambda(r_t)$ decays exponentially when $r_t \to 0$).
In case we do not approximate $\pi(\mathbf{x})$ directly, but we build an emulator of $\log \pi(\mathbf{x})$ or just on the physical model $\mathbf{h}(\mathbf{x})$, it is also possible to show the convergence of the posterior approximation. See, for instance, the error bounds in [77, Theorem 4.2].

5.10.2. A special interesting case for NN-AIS

Here, We focus on NN-AIS. We consider a bounded $\mathcal{X}$ and building $\widehat{\pi}_t$ with a nearest neighbor (NN) approach. In Sect. 5.4.2, we show that the NN emulator at iteration t is given by

$$\widehat{\pi}_t(\mathbf{x}) = \sum_{i=1}^{J_t} \pi(\mathbf{x}_i)\mathbb{I}_{\mathcal{R}_i}(\mathbf{x}) = \sum_{i=1}^{J_t} \pi(\mathbf{x}_i)|\mathcal{R}_i| \left[\frac{1}{|\mathcal{R}_i|}\mathbb{I}_{\mathcal{R}_i}(\mathbf{x})\right], \tag{5.55}$$

$$= \sum_{i=1}^{J_t} v_i \; p_i(\mathbf{x}), \tag{5.56}$$

where $|\mathcal{R}_i|$ is the measure of i-th Voronoi region (see Eq. (5.18) for the definition of $\mathcal{R}_i$), $v_i = \pi(\mathbf{x}_i)|\mathcal{R}_i|$, and $p_i(\mathbf{x}) = \frac{1}{|\mathcal{R}_i|}\mathbb{I}_{\mathcal{R}_i}(\mathbf{x})$ are uniform densities over $\mathcal{R}_i$. Hence, $\widehat{\pi}_t(\mathbf{x})$ is a mixture

of J_t uniform densities where the mixture weight is proportional to ν_i. The normalizing constant of $\widehat{\pi}_t(\mathbf{x})$ is given by

$$c_t = \sum_{i=1}^{J_t} \nu_i = \sum_{i=1}^{J_t} \pi(\mathbf{x}_i)|\mathcal{R}_i|, \tag{5.57}$$

so that the normalized proposal based on the NN emulator is

$$\frac{1}{c_t}\widehat{\pi}_t(\mathbf{x}) = \frac{1}{c_t}\sum_{i=1}^{J_t} \nu_i \; p_i(\mathbf{x}) = \sum_{i=1}^{J_t} \bar{\nu}_i \; p_i(\mathbf{x}),$$

where

$$\bar{\nu}_i = \frac{\nu_i}{c_t} = \frac{\pi(\mathbf{x}_i)|\mathcal{R}_i|}{\sum_{j=1}^{J_t} \pi(\mathbf{x}_j)|\mathcal{R}_j|}, \quad i = 1, ..., N,$$

are also normalized. In order to sample $\frac{1}{c_t}\widehat{\pi}_t(\mathbf{x})$, we would first (i) draw an index i^* from the set $\{1, \ldots, J_t\}$ with probabilities $\bar{\nu}_i = \frac{1}{c_t}\nu_i$ $(i = 1, \ldots, J_t)$, and then (ii) sample from $p_{i^*}(\mathbf{x})$. In practice, we do not know the measures $|\mathcal{R}_i|$ and we are not able to draw samples uniformly in $\mathcal{R}_i$. Hence, we use SIR method to solve the problem drawing from an auxiliary pdf $\bar{q}_{\text{aux}}(\mathbf{x})$ (see 5.10.1), as we have proposed in RADIS. Namely, we resample from the set $\{\mathbf{z}_{t,\ell}\}_{\ell=1}^{L} \sim \bar{q}_{\text{aux}}(\mathbf{x})$ with probabilities proportional to $\gamma_{t,\ell} = \frac{\widehat{\pi}_t(\mathbf{z}_{t,\ell})}{\bar{q}_{\text{aux}}(\mathbf{z}_{t,\ell})}$. Below, we consider the special case that $\bar{q}_{\text{aux}}(\mathbf{x})$ is uniform.

Approximating $\bar{\nu}_i$'s. Let choose an uniform auxiliary density $\bar{q}_{\text{aux}}(\mathbf{x})$, i.e., $\bar{q}_{\text{aux}}(\mathbf{x}) = \frac{1}{|\mathcal{X}|}$ for all $\mathbf{x} \in \mathcal{X}$. We draw $\{\mathbf{z}_{t,\ell}\}_{\ell=1}^{L}$ from the uniform $\bar{q}_{\text{aux}}(\mathbf{x})$. Then, the IS weight associated with the ℓ-th sample is

$$\gamma_{t,\ell} \propto \widehat{\pi}_t(\mathbf{z}_{t,\ell}) = \pi(\mathbf{x}_{k_\ell}),$$

where

$$\mathbf{x}_{k_\ell} = \arg\min_{\mathbf{x}_k \in \mathcal{S}_t} \|\mathbf{x}_k - \mathbf{z}_\ell\|,$$

i.e., $\mathbf{x}_{k_\ell}$ represents the NN of $\mathbf{z}_\ell$ within the set of J_t nodes. Consider now the i-th node $\mathbf{x}_i$. All samples whose NN is $\mathbf{x}_i$ have weight proportional to $\pi(\mathbf{x}_i)$. We denote those samples as the set

$$\mathcal{U}_i = \{\mathbf{z}_{t,\ell} : \; \mathbf{x}_i = \arg\min_{\mathbf{x}_k} \|\mathbf{x}_k - \mathbf{z}_{t,\ell}\|\}. \tag{5.58}$$

The number of samples within $\mathcal{U}_i$ can be written as $|\mathcal{U}_i| = \sum_{\ell=1}^{L} \mathbb{I}(\mathbf{x}_{k_\ell} = \mathbf{x}_i)$. The probability of resampling a $\mathbf{z}_{t,\ell}$ that comes from $\mathcal{U}_i$ is proportional to $|\mathcal{U}_i|\pi(\mathbf{x}_i)$ (since there are $|\mathcal{U}_i|$ samples with weight $\pi(\mathbf{x}_i)$). As $L \to \infty$, by the law of large numbers, we have these probabilities converge to the true ones

$$\frac{|\mathcal{U}_i|\pi(\mathbf{x}_i)}{\sum_{k=1}^{J_t} |\mathcal{U}_k|\pi(\mathbf{x}_k)} \to \frac{|\mathcal{R}_i|\pi(\mathbf{x}_i)}{\sum_{k=1}^{J_t} |\mathcal{R}_k|\pi(\mathbf{x}_k)} = \bar{\nu}_i. \tag{5.59}$$

Rejection sampling. Note also that the samples within $\mathcal{U}_i$ form a particle approximation of the uniform density over $\mathcal{R}_i$. Indeed, taking one sample at random from $\mathcal{U}_i$ corresponds

to applying rejection sampling on $p_i(\mathbf{x})$. In order to see this, consider the rejection sampling setting where $p_i(\mathbf{x})$ is the target probability and $\bar{q}_{\text{aux}}(\mathbf{x}) = \frac{1}{|\mathcal{X}|}$ is the proposal. Note that $\frac{p_i(\mathbf{x})}{\bar{q}_{\text{aux}}(\mathbf{x})} = \frac{|\mathcal{X}|}{|\mathcal{R}_i|}$ for all $\mathbf{x} \in \mathcal{R}_i$, and $\frac{p_i(\mathbf{x})}{\bar{q}_{\text{aux}}(\mathbf{x})} = 0$ for all $\mathbf{x} \notin \mathcal{R}_i$, so $\bar{q}_{\text{aux}}(\mathbf{x})$ is a valid proposal for rejection sampling with rejection constant $M = \frac{|\mathcal{X}|}{|\mathcal{R}_i|}$ [49, Chapter 3]. In rejection sampling, we draw $\mathbf{z} \sim \bar{q}_{\text{aux}}(\mathbf{x})$, $u \sim \mathcal{U}[0, 1]$ and accept $\mathbf{z}$ if

$$u \frac{|\mathcal{X}|}{|\mathcal{R}_i|} \bar{q}_{\text{aux}}(\mathbf{z}) \leq p_i(\mathbf{z}). \tag{5.60}$$

If the condition holds, $\mathbf{z}$ is an independent sample from $p_i(\mathbf{x})$. Otherwise we reject $\mathbf{z}$, draw another candidate $\mathbf{z}$ and so on. Note that, when $\mathbf{z} \in \mathcal{R}_i$, we have

$$u \frac{|\mathcal{X}|}{|\mathcal{R}_i|} \frac{1}{|\mathcal{X}|} \leq \frac{1}{|\mathcal{R}_i|} \iff u \leq 1, \tag{5.61}$$

so we always accept all $\mathbf{z}$'s that are closest to node $\mathbf{x}_i$, becoming i.i.d. samples from $p_i(\mathbf{x})$. Conversely, when $\mathbf{z} \notin \mathcal{R}_i$, we have the condition $u \leq 0$ that never holds, so that we always reject them. Namely, the set $\mathcal{U}_i$ contains i.i.d. samples from $p_i(\mathbf{x})$, that have been obtained by rejection sampling.

Summary. With the particular choice $\bar{q}_{\text{aux}}(\mathbf{x}) = \frac{1}{|\mathcal{X}|}$ for all $\mathbf{x} \in \mathcal{X}$, the SIR approach in NN-AIS is equivalent to (i) estimating by Monte Carlo the mixture probabilities $\bar{\nu}_i$, and (ii) applying rejection sampling to sample uniformly within each Voronoi region $\mathcal{R}_i$.

5.10.3. NN-AIS in unbonded domains

In this section, we recall how to extend the applicability of the nearest neighbor (NN) construction (see Sect. 5.4.2) when the domain $\mathcal{X}$ is unbounded and show how to adapt the support of NN approximation.

NN-AIS with a fixed support in an unbounded domain

Consider again the following mixture proposal,

$$\varphi_t(\mathbf{x}) = \alpha_t \bar{q}_{\texttt{par}}(\mathbf{x}) + (1 - \alpha_t) \frac{1}{c_t} \widehat{\pi}_t(\mathbf{x}), \tag{5.62}$$

where $\alpha_t \in [0, 1]$ for all t, and $\bar{q}_{\texttt{par}}(\mathbf{x})$ is parametric pdf that covers properly the tails of the posterior π. Namely, $\bar{q}_{\texttt{par}}(\mathbf{x})$ is defined in the unbounded domain $\mathcal{X}$ of π, whereas $\widehat{\pi}_t(\mathbf{x})$ is built considering a bounded support $\mathcal{D} \subset \mathcal{X}$, decided in advance by the user. Hence, φ_t is a valid proposal when using NN-AIS with unbounded $\mathcal{X}$. In this simple scenario, $\mathcal{D}$ is fixed and does not vary with the iteration t. However, the information provided by the samples from $\bar{q}_{\texttt{par}}(\mathbf{x})$ can be used to expand the support of $\widehat{\pi}_t$, i.e., such it has an adaptive support, as described below.

Adapting support in NN-AIS

Let $\mathcal{X}$ be unbounded and $\widehat{\pi}_t$ be the surrogate model built with NN. Let $\mathcal{D}_t \subset \mathcal{X}$ denote the compact subset of $\mathcal{X}$ where $\widehat{\pi}_t$ is defined, i.e., $\widehat{\pi}_t$ is zero outside $\mathcal{D}_t$. Note that $\mathcal{D}_t$ depends on t. The set of current nodes $\mathcal{S}_t$ is used to define the boundaries of $\mathcal{D}_t$. One possible way is as follows: Take $\mathcal{D}_t$ as the hyperrectangle whose edges are defined by the maximum and minimum value, in each dimension, of the set $\mathcal{S}_t$, i.e.,

$$\mathcal{D}_t = \{\mathbf{x} \in \mathcal{X} : \min_{\mathbf{s}_{t-1} \in \mathcal{S}_{t-1}} s_{d,t-1} \leq x_d \leq \max_{\mathbf{s}_{t-1} \in \mathcal{S}_{t-1}} s_{d,t-1}, \;\; d = 1, \ldots, d_x\}, \tag{5.63}$$

where x_d denotes the d-th element of $\mathbf{x}$, $\mathbf{s}_{t-1} = [s_{1,t-1}, \ldots, s_{d_x,t-1}] \in \mathcal{S}_{t-1}$ and $\mathcal{S}_{t-1}$ denotes the set of nodes at iteration t. After adding new nodes, we update the bounds of the hyperrectangle. Note that only samples from $\bar{q}_{\texttt{par}}(\mathbf{x})$ that fall outside $\mathcal{D}_t$ will expand it. Note that, the size of $\mathcal{D}_t$ is always increasing but controlled by the tail of $\bar{\pi}$. Indeed, the candidate samples drawn in the tails of $\bar{\pi}$ will have very low values of π, so that the probability of sampling those regions will be negligible (then these regions will be never used). In the case we use a uniform $\bar{q}_{\text{aux}}(\mathbf{x})$ in $\mathcal{D}_t$ to sample $\widehat{\pi}_t$, note that $\bar{q}_{\text{aux}}(\mathbf{x})$ actually depends on t and is changing at every iteration whenever $\mathcal{D}_t$ changes.

Bibliography

[1] Ö. D. Akyildiz and J. Míguez. Convergence rates for optimised adaptive importance samplers. *arXiv preprint arXiv:1903.12044*, 2019.

[2] M. Balesdent, J. Morio, and J. Marzat. Kriging-based adaptive importance sampling algorithms for rare event estimation. *Structural Safety*, 44:1–10, 2013.

[3] S. C. C Barros et al. WASP-113b and WASP-114b, two inflated hot Jupiters with contrasting densities. *Astronomy and Aastrophysics*, 593:A113, 2016.

[4] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric. Adaptive importance sampling: the past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017.

[5] Daniel Busby. Hierarchical adaptive experimental design for Gaussian process emulators. *Reliability Engineering & System Safety*, 94(7):1183–1193, 2009.

[6] T. Butler, L. Graham, S. Mattis, and S. Walsh. A measure-theoretic interpretation of sample based numerical integration with applications to inverse and prediction problems under uncertainty. *SIAM Journal on Scientific Computing*, 39(5):A2072–A2098, 2017.

[7] G. Camps-Valls, D. Sejdinovic, J. Runge, and M. Reichstein. A perspective on Gaussian processes for Earth observation. *National Science Review*, 6:616–618, 2019.

[8] Gustau Camps-Valls, Daniel Svendsen, Luca Martino, Jordi Munoz-Mari, Valero Laparra, Manuel Campos-Taberner, and David Luengo. Physics-aware Gaussian processes in remote sensing. *Applied Soft Computing*, 68:69–82, Jul 2018.

[9] O. Cappé, R. Douc, A. Guillin, J. M. Marin, and C. P. Robert. Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18:447–459, 2008.

[10] O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.

[11] J. A. Christen and C. Fox. Markov Chain Monte Carlo using an approximation. *Journal of Computational and Graphical statistics*, 14(4):795–810, 2005.

[12] E. Cleary, A. Garbuno-Inigo, S. Lan, T. Schneider, and A. M. Stuart. Calibrate, emulate, sample. *arXiv:2001.03689*, 2020.

[13] P. R. Conrad, Y. M. Marzouk, N. S. Pillai, and A. Smith. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516):1591–1607, 2016.

[14] J. M. Cornuet, J. M. Marin, A. Mira, and C. P. Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, December 2012.

[15] L. Devroye, L. Györfi, G. Lugosi, and H. Walk. On the measure of Voronoi cells. *Journal of Applied Probability*, 54(2):394–408, 2017.

[16] X. Dong, D. Thanou, M. Rabbat, and P. Frossard. Learning graphs from data: A signal representation perspective. *IEEE Signal Processing Magazine*, 36(3):44–63, 2019.

[17] V. Dubourg, B. Sudret, and F. Deheeger. Metamodel-based importance sampling for structural reliability analysis. *Probabilistic Engineering Mechanics*, 33:47–57, 2013.

[18] Y. El-Laham, P. M. Djurić, and M. F. Bugallo. A variational adaptive population importance sampler. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5052–5056. IEEE, 2019.

[19] V. Elvira, L. Martino, and P. Closas. Importance Gaussian Quadrature. *arXiv:2001.03090*, 2020.

[20] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Improving population Monte Carlo: Alternative weighting and resampling schemes. *Signal Processing*, 131:77–91, 2017.

[21] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Generalized multiple importance sampling. *Statistical Science*, 34(1):129–155, 2019.

[22] J. Felip, N. Ahuja, and O. Tickoo. Tree pyramidal adaptive importance sampling. *arXiv preprint arXiv:1912.08434*, 2019.

[23] T. Foster, C. L. Lei, M. Robinson, D. Gavaghan, and B. Lambert. Model evidence with fast tree based quadrature. *arXiv preprint arXiv:2005.11300*, 2020.

[24] J. H. Friedman and M. H. Wright. A nested partitioning procedure for numerical multiple integration. *ACM Transactions on Mathematical Software (TOMS)*, 7(1):76–92, 1981.

[25] A. Gelman and X.-L. Meng. *Applied Bayesian modeling and causal inference from incomplete-data perspectives*. John Wiley & Sons, 2004.

[26] W. R. Gilks, N. G. Best, and K. K. C. Tan. Adaptive Rejection Metropolis Sampling within Gibbs Sampling. *Applied Statistics*, 44(4):455–472, 1995.

[27] W. R. Gilks and P. Wild. Adaptive Rejection Sampling for Gibbs Sampling. *Applied Statistics*, 41(2):337–348, 1992.

[28] G. H. Givens and J. A. Hoeting. *Computational statistics*, volume 703. John Wiley & Sons, 2012.

[29] D. Görür and Y. W. Teh. Concave convex adaptive rejection sampling. *Journal of Computational and Graphical Statistics*, 20(3):670–691, 2011.

[30] Philip C. Gregory. Bayesian re-analysis of the Gliese 581 exoplanet system. *Monthly Notices of the Royal Astronomical Society*, 415(3):2523–2545, August 2011.

[31] T. E. Hanson, J. V. D. Monteiro, and A. Jara. The Polya tree sampler: Toward efficient and automatic independent Metropolis–Hastings proposals. *Journal of Computational and Graphical Statistics*, 20(1):41–62, 2011.

[32] W. Hörmann. A rejection technique for sampling from T-concave distributions. *ACM Transactions on Mathematical Software*, 21(2):182–193, 1995.

[33] S. Jacquemoud, W. Verhoef, F. Baret, C. Bacour, P.J. Zarco-Tejada, G.P. Asner, C. François, and S.L. Ustin. PROSPECT+ SAIL models: A review of use for vegetation characterization. *Remote sensing of environment*, 113:S56–S66, 2009.

[34] M. Kennedy. Bayesian quadrature with non-normal approximating functions. *Statistics and Computing*, 8(4):365–375, 1998.

[35] M.C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 63(3):425–450, 2001.

[36] G. P. Lepage. A new algorithm for adaptive multidimensional integration. *Journal of Computational Physics*, 27(2):192–203, 1978.

[37] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2004.

[38] J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[39] F. Llorente, L. Martino, D. Delgado, and J. Lopez-Santiago. Marginal likelihood computation for model selection and hypothesis testing: an extensive review. *viXra:2001.0052*, 2019.

[40] F. Llorente, L. Martino, V. Elvira, D. Delgado, and J. Lopez-Santiago. Adaptive quadrature schemes for Bayesian inference via active learning. *IEEE Access*, 8:208462–208483, 2020.

[41] X. Lu, T. Rainforth, Y. Zhou, J.-W. van de Meent, and Y. W. Teh. On exploration, exploitation and learning in adaptive importance sampling. *arXiv preprint arXiv:1810.13296*, 2018.

[42] G. Marsaglia and W. W. Tsang. The Ziggurat method for generating random variables. *Journal of Statistical Software*, 8(5):1–7, 2000.

[43] L. Martino. Parsimonious adaptive rejection sampling. *Electronics Letters*, 53(16):1115–1117, 2017.

[44] L. Martino, R. Casarin, F. Leisen, and D. Luengo. Adaptive independent sticky MCMC algorithms. *EURASIP Journal on Advances in Signal Processing*, 2018(1):5, 2018.

[45] L. Martino, V. Elvira, and G. Camps-Valls. Group Importance Sampling for particle filtering and MCMC. *Digital Signal Processing*, 82:133–151, 2018.

[46] L. Martino, V. Elvira, and F. Louzada. Effective sample size for importance sampling based on discrepancy measures. *Signal Processing*, 131:386 – 401, 2017.

[47] L. Martino, V. Elvira, D. Luengo, and J. Corander. An adaptive population importance sampler: Learning from the uncertanity. *IEEE Transactions on Signal Processing*, 63(16):4422–4437, 2015.

[48] L. Martino, V. Elvira, D. Luengo, and J. Corander. Layered adaptive importance sampling. *Statistics and Computing*, 27(3):599–623, 2017.

[49] L. Martino, D. Luengo, and J. Miguez. *Independent Random Sampling methods*. Springer, 2018.

[50] L. Martino, D. Luengo, and J. Míguez. Independent random sampling methods. *Springer*, 2018.

[51] L. Martino and J. Read. Joint introduction to Gaussian Processes and Relevance Vector Machines with connections to Kalman filtering and other kernel smoothers. *arXiv:2009.09217*, 2020.

[52] L. Martino, J. Read, V. Elvira, and F. Louzada. Cooperative parallel particle filters for on-line model selection and applications to urban mobility. *Digital Signal Processing*, 60:172–185, 2017.

[53] L. Martino, J. Read, and D. Luengo. Independent doubly adaptive rejection metropolis sampling within gibbs sampling. *IEEE Transactions on Signal Processing*, 63(12):3123–3138, 2015.

[54] L. Martino, H. Yang, D. Luengo, J. Kanniainen, and J. Corander. A fast universal self-tuned sampler within Gibbs sampling. *Digital Signal Processing*, 47:68 – 83, 2015.

[55] R. Meyer, B. Cai, and F. Perron. Adaptive rejection Metropolis sampling using Lagrange interpolation polynomials of degree 2. *Computational Statistics and Data Analysis*, 52(7):3408–3423, March 2008.

[56] C. Musso, N. Oudjane, and F. Le Gland. Improving regularised particle filters. In *Doucet A., de Freitas N., Gordon N. (eds) Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science. Springer, New York*, pages 247–271, 2001.

[57] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

[58] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial Mathematics, 1992.

[59] A. O'Hagan. *Bayesian Inference*, volume 2B. Kendall's Advanced Theory of Statistics. Arnold, London, UK, 1994.

[60] A. O'Hagan. *Bayesian Inference, volume 2B of Kendall's Advanced Theory of Statistics*. Arnold, London, United Kingdom, 1994.

[61] A. O'Hagan. Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering and System Safety*, 91(10-11):1290–1300, 2006.

[62] M. Osborne, R. Garnett, Z. Ghahramani, D. K. Duvenaud, S. J. Roberts, and C. E. Rasmussen. Active learning of model evidence using Bayesian quadrature. In *Advances in neural information processing systems*, pages 46–54, 2012.

[63] A. Owen and Y. Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.

[64] Anthony O'Hagan. Probabilistic uncertainty specification: Overview, elaboration techniques and their application to a mechanistic model of carbon flux. *Environmental Modelling & Software*, 36(0):35 – 48, 2012. Thematic issue on Expert Opinion in Environmental Modelling and Management.

[65] W. H. Press and G. R. Farrar. Recursive stratified sampling for multidimensional Monte Carlo integration. *Computers in Physics*, 4(2):190–195, 1990.

[66] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C++ : the art of scientific computing*. Springer, 2002.

[67] C. E. Rasmussen and Z. Ghahramani. Bayesian Monte Carlo. *Advances in neural information processing systems*, pages 505–512, 2003.

[68] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, New York, 2006.

[69] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[70] Saman Razavi, Bryan A Tolson, and Donald H Burn. Review of surrogate modeling in water resources. *Water Resources Research*, 48(7), 2012.

[71] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[72] C. P. Robert and W. Changye. Markov Chain Monte Carlo Methods, a survey with some frequent misunderstandings. *arXiv preprint arXiv:2001.06249*, 2020.

[73] D. B. Rubin. Using the SIR algorithm to simulate posterior distributions. *in Bayesian Statistics 3, ads Bernardo, Degroot, Lindley, and Smith. Oxford University Press, Oxford, 1988.*, 1988.

[74] Claude J Schmit and Jonathan R Pritchard. Emulation of reionization simulations for bayesian inference of astrophysics parameters using neural networks. *Monthly Notices of the Royal Astronomical Society*, 475(1):1213–1223, 2018.

[75] A. F..M. Smith and A. E. Gelfand. Bayesian statistics without tears: a sampling–resampling perspective. *The American Statistician*, 46(2):84–88, 1992.

[76] R. J. Steele, A. E. Raftery, and M. J. Emond. Computing normalizing constants for finite mixture models via incremental mixture importance sampling (IMIS). *Journal of Computational and Graphical Statistics*, 15(3):712–734, 2006.

[77] A. Stuart and A. Teckentrup. Posterior consistency for Gaussian process approximations of Bayesian posterior distributions. *Mathematics of Computation*, 87(310):721–753, 2018.

[78] D. H. Svendsen, L. Martino, and G. Camps-Valls. Active emulation of computer codes with gaussian processes - application to remote sensing. *Pattern Recognition*, 100:107103, 2020.

[79] D. H. Svendsen, L. Martino, and G. Camps-Valls. Active emulation of computer codes with Gaussian processes–Application to remote sensing. *Pattern Recognition*, 100:107103, 2020.

[80] Trifon Trifonov, Stephan Stock, Thomas Henning, Sabine Reffert, Martin Kürster, Man Hoi Lee, Bertram Bitsch, R. Paul Butler, and Steven S. Vogt. Two Jovian Planets around the Giant Star HD 202696: A Growing Population of Packed Massive Planetary Pairs around Massive Stars? *The Astronomical Journal*, 157(3):93, March 2019.

[81] E. Veach and L. J. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, 1995.

[82] H. Ying, K. Mao, and K. Mosegaard. Moving Target Monte Carlo. *arXiv preprint arXiv:2003.04873*, 2020.

[83] P. Zhang. Nonparametric Importance Sampling. *Journal of the American Statistical Association*, 91(435):1245–1253, 1996.

6. A SURVEY OF MONTE CARLO METHODS FOR NOISY AND COSTLY DENSITIES WITH APPLICATION TO REINFORCEMENT LEARNING

In *arXiv preprint*, arXiv:2108.00490 (2021)

F. Llorente*, L. Martino⋆, J. Read†, D. Delgado*

* Universidad Carlos III de Madrid, Leganés (Spain).

⋆ Universidad Rey Juan Carlos, Fuenlabrada (Spain).

† École Polytechnique, Palaiseau (France).

Abstract

This survey gives an overview of Monte Carlo methodologies using surrogate models, for dealing with densities which are intractable, costly, and/or noisy. This type of problem can be found in numerous real-world scenarios, including stochastic optimization and reinforcement learning, where each evaluation of a density function may incur some computationally-expensive or even physical (real-world activity) cost, likely to give different results each time. The surrogate model does not incur this cost, but there are important trade-offs and considerations involved in the choice and design of such methodologies. We classify the different methodologies into three main classes and describe specific instances of algorithms under a unified notation. A modular scheme which encompasses the considered methods is also presented. A range of application scenarios is discussed, with special attention to the likelihood-free setting and reinforcement learning. Several numerical comparisons are also provided.

Keywords: Noisy Monte Carlo; Intractable Likelihoods; Approximate Bayesian Computation; Pseudo Marginal Metropolis; Surrogate models.

6.1. Introduction

Bayesian methods and their implementations by means of sophisticated Monte Carlo techniques, such as Markov chain Monte Carlo (MCMC) and importance sampling (IS) schemes, have become very popular [56, 39]. In the last years, there is a broad interest in performing Bayesian inference in models where the posterior probability density function (pdf) is analytically *intractable*, and/or *costly* to evaluate, and/or its evaluation is *noisy*. Namely, there are several practical situations where the posterior distribution cannot be evaluated pointwise or its evaluation is expensive [25, 2, 49, 42]. Such models occur in a wide range of applications including spatial statistics, social network analysis, statistical genetics, finance, etc. For instance, **(a)** for the use of massive datasets where the likelihood consists of a product of a large number of terms [10], or **(b)** for the existence of

a large number of latent variables that we should marginalize out (hence, the posterior pdf can be obtained only solving a high dimensional integral) [5]. Moreover, another scenario is **(c)** when a piece of likelihood function is analytically unknown and it should be approximated [49, 25]. The intractable likelihood models arising from, for example, Markov random fields, such as those found in spatial statistics and network analysis [58]. In many settings, **(d)** the likelihood function is induced by a complex stochastic computer model which is costly to evaluate pointwise [40]. **(e)** In other application fields, such as reinforcement learning, a target function (usually a policy) cannot be exactly evaluated neither quickly nor precisely, since such an evaluation corresponds to interaction with an environment (possibly in the real world) which is inherently lengthy to obtain and susceptible to contamination by noise perturbation. Hence, the evaluation is obtained with a certain degree of uncertainty [17].

Noisy computational schemes. The solutions proposed in the literature to performing the inference in the scenarios **(a)**-**(b)**-**(c)** above, have been carried out using Monte Carlo algorithms which often consider noisy evaluations of the target density [10, 5, 3, 49]. A natural approach in these cases is to replace the intractable/costly model with an approximation (or with a pointwise estimation in the case of a noisy model). Thus, the corresponding Monte Carlo schemes also involve the use a *surrogate model* via regression techniques. Furthermore, in the scenario **(d)**, if it is possible to draw artificial data according the observation model, sometimes is preferable to generate fake data (given some parameters) and to measure the discrepancy between the generated data and the actual data, instead of evaluating the costly likelihood function [11, 42]. This approach is known as Approximate Bayesian Computation (ABC). This area has generated much activity in the literature (see, e.g., [55]). The discrepancy measure plays the role a surrogate model and, due to the stochastic generation of the artificial data, it also adds uncertainty (i.e., as a noise perturbation) in the internal evaluations within the ABC-Monte Carlo methods [42]. Finally, The last scenario **(e)** is intrinsically noisy, so that it also requires specific computational solutions.

The three different cases above, *intractable*, *costly* and *noisy* evaluations of a posterior distribution can appear and/or can be addressed separately [40, 2, 42]. In all of these cases, a surrogate model can accelerate the Monte Carlo method or approximate the posterior distribution [18, 50, 59, 34]. As described above, these cases also appear jointly in real-world applications (specially, if we consider the algorithms designed to address those issues): 'intractable and costly', 'intractable and noisy', or 'costly and noisy' posterior evaluations, etc. The challenge posed by these contexts has led to the development of recent theoretical and methodological advances in the literature. Furthermore, surrogate models have been considered as an alternative to Monte Carlo for approximating complicated integrals. Here, the surrogate is substituted *directly* into the integral of interest, instead of the original density (e.g., a posterior). A cubature rule is subsequently obtained, which makes a more efficient use of the posterior evaluations [14, 36, 37].

Contribution. In this work, we provide a survey of methods which use surrogate models *within* Monte Carlo algorithms for dealing with noisy and costly posteriors. Some of them

have been introduced only in the context of expensive posteriors [18]. Other schemes have been designed only for improving the efficiency of the Monte Carlo methods considering a more sophisticated proposal density (see for instance, [44, 40]). However, all of them can be applied also in a noisy scenario. In Sections 6.2 and 6.3, we provide a general joint framework which encompasses most of the techniques in the literature. We introduce the vanilla schemes for noisy MH method (well studied in the literature, e.g., [5, 22]) and also of a noisy IS scheme (which also has been studied in the literature in works such as [26, 63]). We focus mainly on the static batch scenario for MCMC and IS algorithms. However, most of the results presented in this work can be extended to the sequential framework (consider, e.g., the recent work of [13]).
We classify the studied techniques in different families, and provide several explanatory tables and figures. More specifically, we divide the algorithms in the literature in three broad classes: 1) two-stage, 2) iterative refinement, and 3) exact. In Section 6.4, we also provide detailed descriptions of specific examples of algorithms. For instance, we provide a generic description of Metropolis-Hastings (MH) schemes on an iterative surrogate. The *moving target MH* algorithm is a specific example of this [67]. Then, we describe some specific implementation of the so-called *Delayed Acceptance MH* (DA-MH) methods [9]. We also introduce *Noisy Deep Importance Sampling* (N-DIS) which is a noisy version of the Deep IS method in [40]. The range of application of the methods described above is also discussed in Section 6.5. More specifically, we give a detailed description of two scenarios: the likelihood-free approach in Section 6.5.1, and the reinforcement learning (RL) setting in Section 6.5.2 [62, 32]. We test the presented algorithms in different numerical experiments in Section 6.6. The application to a benchmark RL problem, the double cart-pole system [31], is given in Section 6.6.3. Finally, we conclude with brief discussion in Section 6.7.

6.2. General framework

Let us assume that our goal is the study of the unnormalized density $p(\boldsymbol{\theta})$, $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^d$ using Monte Carlo methods. For instance, $p(\boldsymbol{\theta})$ may represent a posterior density in a Bayesian inference problem. There are two problems: **(P1)** for any $\boldsymbol{\theta}$, we cannot evaluate $p(\boldsymbol{\theta})$ exactly, but we only have access to a related noisy realization, and **(P2)** obtaining such a noisy realization is expensive. Typically, this occurs in applications where the function of interest $p(\boldsymbol{\theta})$ is intractable or expensive to evaluate. More specifically, in many practical cases, we have access to a noisy realization related to $p(\boldsymbol{\theta})$, i.e.,

$$\widetilde{m}(\boldsymbol{\theta}) = H(p(\boldsymbol{\theta}), \epsilon), \tag{6.1}$$

where H is a non-linear transformation involving $p(\boldsymbol{\theta})$ and ϵ, that is some noise perturbation. Thus, for a given $\boldsymbol{\theta}$, $\widetilde{m}(\boldsymbol{\theta})$ is a random variable with

$$\mathbb{E}[\widetilde{m}(\boldsymbol{\theta})] = m(\boldsymbol{\theta}), \quad \texttt{var}[\widetilde{m}(\boldsymbol{\theta})] = s^2(\boldsymbol{\theta}), \tag{6.2}$$

for some *mean function*, $m(\boldsymbol{\theta})$, and *variance function*, $s^2(\boldsymbol{\theta})$. Some examples of noisy models with the corresponding mean and variance functions are given in Appendix 6.8.3. The unbiased case, $m(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$, appears naturally in some applications, or it is often assumed as a pre-established condition by the authors. In some other scenarios, the noisy realizations are known to be unbiased estimates of some transformation of $p(\boldsymbol{\theta})$, e.g., of $\log p(\boldsymbol{\theta})$. This situation can be encompassed by the following special case. If we consider an additive perturbation,

$$\widetilde{m}(\boldsymbol{\theta}) = G\left(p(\boldsymbol{\theta})\right) + \epsilon, \qquad \text{with } E[\epsilon] = 0, \tag{6.3}$$

we have $m(\boldsymbol{\theta}) = G\left(p(\boldsymbol{\theta})\right)$. If $G(\cdot) : \mathbb{R} \to \mathbb{R}$ is known and invertible, we have $p(\boldsymbol{\theta}) = G^{-1}\left(m(\boldsymbol{\theta})\right)$.

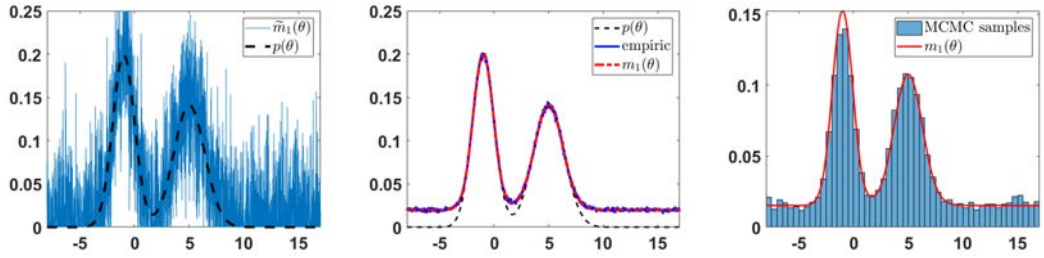
Remark 1. *Generally, transforming $\widetilde{m}(\boldsymbol{\theta})$ into an unbiased realization of $p(\boldsymbol{\theta})$ is not straightforward, since $\mathbb{E}[G^{-1}(\widetilde{m}(\boldsymbol{\theta}))] \neq p(\boldsymbol{\theta})$. However, there are cases such as $\widetilde{m}(\boldsymbol{\theta}) = \log p(\boldsymbol{\theta}) + \epsilon$, where we can take $\widetilde{p}(\boldsymbol{\theta}) = e^{\widetilde{m}(\boldsymbol{\theta})}$ which fulfills $\mathbb{E}[\widetilde{p}(\boldsymbol{\theta})] \propto p(\boldsymbol{\theta})$ [34, 23].*

In a general case, we can state that $m(\boldsymbol{\theta})$ always contains statistical information related to $p(\boldsymbol{\theta})$. The subsequent use of $m(\boldsymbol{\theta})$ depends on the specific application. In some settings, it is also possible to control the noise level, by adding/removing data to the mini-batches (e.g., in he context of Big Data) or interacting with an environment over longer/shorter periods of time (e.g., in reinforcement learning). See the Section 6.5 and, in particular, Section 6.5.2 for more details.
Different noise models have different behaviours of the variance function $s^2(\boldsymbol{\theta})$. For instance, an additive Gaussian noise with standard deviation $\sigma_\epsilon(\boldsymbol{\theta})$ is usually assumed in the noisy optimization literature [7, 35]. The location dependence of $\sigma_\epsilon(\boldsymbol{\theta})$ give rise to different behaviors. Some authors consider $\sigma_\epsilon(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta})$, i.e., noise strength proportional to function values, which is interesting in practice [7, 35, 48] (see also Figure 6.2). An illustrative one-dimensional example is provided below, showing a bimodal $p(\boldsymbol{\theta})$ perturbed with two different noises, and the corresponding $m(\boldsymbol{\theta})$.

Illustrative example in 1D. As an illustration, let us consider the one-dimensional density $p(\theta) = \frac{1}{2}\mathcal{N}(\theta; -1, 1) + \frac{1}{2}\mathcal{N}(\theta; 5, 2)$, restricted in the finite domain $[-8, 17]$, and two noisy versions

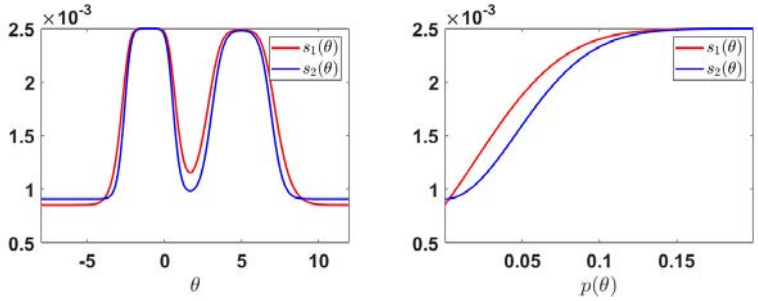
$$\widetilde{m}_1(\theta) = \max(0, p(\theta) + \epsilon), \text{ and } \widetilde{m}_2(\theta) = |p(\theta) + \epsilon|,$$

where $\epsilon \sim \mathcal{N}(0, 0.05^2)$. Namely, $\widetilde{m}_i(\theta)$, $i = 1, 2$, correspond to rectified Gaussian and folded Gaussian random variables, respectively (for any θ). In Figure 6.1-(a), we show one realization of $\widetilde{m}_1(\theta)$. In Figure 6.1-(b), we show the average of $\widetilde{m}_1(\theta)$ (empirically and theoretically). In Figure 6.1-(c), we show the histogram of samples obtained by running a (pseudo-marginal) MH algorithm on $\widetilde{m}_1(\boldsymbol{\theta})$. In these cases, the expected values do not coincide with $p(\theta)$, i.e., $m_i(\theta) \neq p(\theta)$. Analytical expressions of $m_i(\theta)$, as well as $s_i^2(\theta)$, can be obtained as shown in App. 6.8.3. The variance behaviors are depicted in Figures 6.2(a)–(b).



(a) $\widetilde{m}_1(\theta)$ (b) $m_1(\theta)$ and $p(\boldsymbol{\theta})$ (c) histogram of samples

Figure 6.1: **(a)** The target pdf $p(\theta)$ and a realization of $\widetilde{m}_1(\theta) = \max(0, p(\theta)+\epsilon)$. **(b)** Again the target pdf $p(\theta)$ (dashed line), the mean function $m(\theta) = \mathbb{E}[\widetilde{m}(\theta)]$ and its empirical approximation averaging several realizations. **(c)** Histogram of the samples generated by a noisy MCMC scheme.



(a) $s_i(\theta)$ vs θ (b) $s_i(\theta)$ vs $p(\theta)$

Figure 6.2: Behavior of the variance $s_i^2(\theta)$ in both models. **(a)** On the left: Plots of $\sqrt{s_i^2(\theta)}$ versus θ for $i = 1, 2$. **(b)** Plots of $\sqrt{s_i^2(\theta)}$ versus $p(\theta)$ for $i = 1, 2$.

6.2.1. Vanilla schemes for Noisy MH and noisy IS

In this Section, we present two basic Monte Carlo algorithms working with noisy realizations $\widetilde{m}(\boldsymbol{\theta})$.

Noisy MH. The standard MH algorithm produces correlated samples from a target distribution $p(\boldsymbol{\theta})$ by sampling candidates from a proposal density which are either rejected or accepted according to a suitable probability. The evaluation of the target density $p(\boldsymbol{\theta})$ is required at each iteration. A noisy version of this algorithm is obtained when we substitute the evaluations of $p(\boldsymbol{\theta})$ (at the candidate points) with a realization of the random variable $\widetilde{m}(\boldsymbol{\theta})$. The algorithm is shown in Table 6.1. If a different noisy realization $\widetilde{m}(\boldsymbol{\theta}_{t-1})$ is obtained at each iteration, this algorithms is called *Monte Carlo-within-Metropolis* technique [46]. On the contrary, if it is recycled from the previous iteration, the algorithm is called *pseudo-marginal* MH (PM-MH) algorithm [5]. The latter approach ensures the algorithm is "exact" (see Theorem 1).

Noisy IS. In a standard IS scheme, a set of samples is drawn from a proposal density $q(\boldsymbol{\theta})$. Then each sample is weighted according to the ratio $\frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}$. Like in the MH case, a noisy

version of importance sampling can be obtained when we substitute the evaluations of $p(\boldsymbol{\theta})$ with noisy realizations of $\widetilde{m}(\boldsymbol{\theta})$. See Table 6.2.

Theorem 1. *Under certain conditions, the estimators constructed from the output of noisy MH algorithm and noisy IS converge to expectations under $m(\boldsymbol{\theta})$.*

Proof. For the MH algorithm, see [5, 6] and App. 6.8.1. For noisy IS, see App. 6.8.2. □

Theorem 2. *The noisy estimators derived from noisy MH and noisy IS have higher variance than their non-noisy counterparts.*

Proof. For the MH algorithm, see [6] For noisy IS, see App. 6.8.2. □

Table 6.1: Noisy Metropolis-Hastings (N-MH) algorithms

1. **Inputs:** Initial state $\boldsymbol{\theta}_0$ and realization $\widetilde{m}(\boldsymbol{\theta}_0)$.
2. For $t = 1, \ldots, T$:
   (a) Sample $\boldsymbol{\theta}_{\text{prop}} \sim \varphi(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$ and obtain realization $\widetilde{m}_{\text{now}} = \widetilde{m}(\boldsymbol{\theta}_{\text{prop}})$.
   (b) In the so-called *pseudo-marginal MH* (PM-MH) set $\widetilde{m}_{\text{bef}} = \widetilde{m}(\boldsymbol{\theta}_{t-1})$; otherwise, in the so-called *Monte Carlo-within-MH*, obtain a new realization $\widetilde{m}$ at $\boldsymbol{\theta}_{t-1}$ and set $\widetilde{m}_{\text{bef}} = \widetilde{m}(\boldsymbol{\theta}_{t-1})$.
   (c) With probability
   $$\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_{\text{prop}}) = \min\left\{1, \frac{\widetilde{m}_{\text{now}}\ \varphi(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}_{\text{prop}})}{\widetilde{m}_{\text{bef}}\ \varphi(\boldsymbol{\theta}_{\text{prop}}|\boldsymbol{\theta}_{t-1})}\right\}, \tag{6.4}$$
   accept $\boldsymbol{\theta}_{\text{prop}}$, i.e., set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{\text{prop}}$. Otherwise, reject $\boldsymbol{\theta}_{\text{prop}}$, i.e., set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$.
3 **Outputs:** the chain $\{\boldsymbol{\theta}_t\}_{t=1}^T$.

Table 6.2: Noisy importance sampling algorithm

1. **Inputs:** Proposal distribution $q(\boldsymbol{\theta})$.
2. For $n = 1, \ldots, N$:
   (a) Sample $\boldsymbol{\theta}_n \sim q(\boldsymbol{\theta})$ and obtain realization $\widetilde{m}(\boldsymbol{\theta}_n)$.
   (b) Compute
   $$w_n = \frac{\widetilde{m}(\boldsymbol{\theta}_n)}{q(\boldsymbol{\theta}_n)} \tag{6.5}$$
3 Compute normalized weights: $\bar{w}_n = \frac{w_n}{\sum_{j=1}^N w_j},\ j = 1, \ldots, N$.
4 **Outputs:** the weighted samples $\{\boldsymbol{\theta}_n, \bar{w}_n\}_{n=1}^N$.

6.2.2. Accelerating and denoising by surrogates

The vanilla schemes described above can be improved by building surrogate regression models $\widehat{m}(\boldsymbol{\theta})$ from the noisy realizations. More specifically, considering the set of J observed points $\{\boldsymbol{\theta}_i, \widetilde{m}(\boldsymbol{\theta}_i)\}_{i=1}^J$, we apply a regression model for obtaining $\widehat{m}(\boldsymbol{\theta})$. We assume to use a surrogate regression model such that $\widehat{m}(\boldsymbol{\theta})$ converges to $m(\boldsymbol{\theta})$ as $J \to \infty$ The locations of the nodes can be chosen appropriately for ensuring the convergence when $J \to \infty$, under mild conditions. The accelerated schemes are obtained replacing $\widetilde{m}(\boldsymbol{\theta})$ with $\widehat{m}(\boldsymbol{\theta})$ in the Tables 6.1 and 6.2 above. Then, the resulting algorithms target $\widehat{m}(\boldsymbol{\theta})$.

Remark 2. *A necessary condition is that the construction of $\widehat{m}(\boldsymbol{\theta})$ must be strictly positive, $\widehat{m}(\boldsymbol{\theta}) > 0$, for all $\boldsymbol{\theta}$ where $m(\boldsymbol{\theta}) > 0$.*

Remark 3. *Note that, if the transformation in Eq.* (6.3) *is known, we can undo it in order to obtain $\widehat{p}(\boldsymbol{\theta}) = G^{-1}(\widehat{m}(\boldsymbol{\theta}))$ and use it within the algorithms, which will target $\widehat{p}(\boldsymbol{\theta})$, instead of $\widehat{m}(\boldsymbol{\theta})$.*

Remark 4. *Even if the transformation G is known in Eq.* (6.3)*, and we can obtain $\widehat{p}(\boldsymbol{\theta}) = G^{-1}(\widehat{m}(\boldsymbol{\theta}))$, in general we have $\mathbb{E}[\widehat{p}(\boldsymbol{\theta})] \not\propto p(\boldsymbol{\theta})$. One exception is the case $G(p(\boldsymbol{\theta})) = \log p(\boldsymbol{\theta})$, which implies $\mathbb{E}[\widehat{p}(\boldsymbol{\theta})] = \mathbb{E}[G^{-1}(\widetilde{m}(\boldsymbol{\theta}))] = \mathbb{E}[e^{\epsilon} p(\boldsymbol{\theta})] = \mathbb{E}[e^{\epsilon}] p(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta})$.*

Clearly, the selection of the design nodes $\{\boldsymbol{\theta}_i, \widetilde{m}(\boldsymbol{\theta}_i)\}_{i=1}^J$ is a very important point. In the Monte Carlo literature, strategies for obtaining the set of design nodes are, for instance, running a pilot MCMC run [23], applying Bayesian experimental design algorithms [34, 61], space-filling heuristics [18, 41], or optimization [12]. In iterative refinement, the path of the chain can also be used to update the surrogate, either *directly* by including some of the states of the chain [67], or *indirectly* by guiding the search of design points with other techniques [18].

Note that the use of a surrogate is beneficial for working with both costly and noisy target pdfs. In the following, we review different MCMC and IS approaches that can deal with noisy and expensive target distributions. Some of these methods have been originally proposed only for the expensive or just noisy case (i.e., in a more restricted range of application), but they can address the complete problem considered in this work. A schematic summary of the main notation of the work is given below.

Density	**Noisy realization**	**Surrogate**
$p(\boldsymbol{\theta})$	$\widetilde{m}(\boldsymbol{\theta}) = H(p(\boldsymbol{\theta}), \epsilon)$	$\widehat{m}(\boldsymbol{\theta})$

More generally, in the MCMC context, approximations of the whole acceptance ratio can be built and used instead of the true one. Properties of these "approximate" chains are studied in [2].

6.3. Overview and generic scheme

In this Section, we present a general scheme that combines Monte Carlo with the use of surrogates, which encompasses most of the methods proposed in the literature for costly or noisy target pdfs. Moreover, we distinguish three main classes of methods: **(C1)** *two-stage*, **(C2)** *iterative refinement*, and **(C3)** *exact* schemes. Below, we provide a brief description of each of them.
A graphical representation of the generic scheme is given in Figure 6.3, that is composed of a series of blocks. Each approach in the literature is formed by a different combination of blocks (e.g., see Table 6.4). The three main classes C1, C2, C3 have in common the Block 2, i.e., performing one or more Monte Carlo iterations (e.g., MH or IS) with respect to (w.r.t.) the surrogate $\widehat{m}(\boldsymbol{\theta})$ instead of $\widetilde{m}(\boldsymbol{\theta})$.

Remark 5. *Note that this block can be viewed as sampling from a non-parametric proposal. Furthermore, the application of Monte Carlo in Block 2 could be substituted with a direct sampling of the surrogate when it is possible [44].*

Blocks 1 and 3 refer to the two possible strategies for building the surrogate. The former considers an offline construction, that is totally independent of the Monte Carlo algorithm that will be run afterwards. The latter construction aims to build the surrogate online, i.e., during the Monte Carlo iterations. Lastly, Block 4 refers to making a correction for the fact that we are working w.r.t. $\widehat{m}(\boldsymbol{\theta})$, and ultimately implies obtaining a noisy realization $\widetilde{m}(\boldsymbol{\theta})$. The schemes are presented in increasing order of complexity.

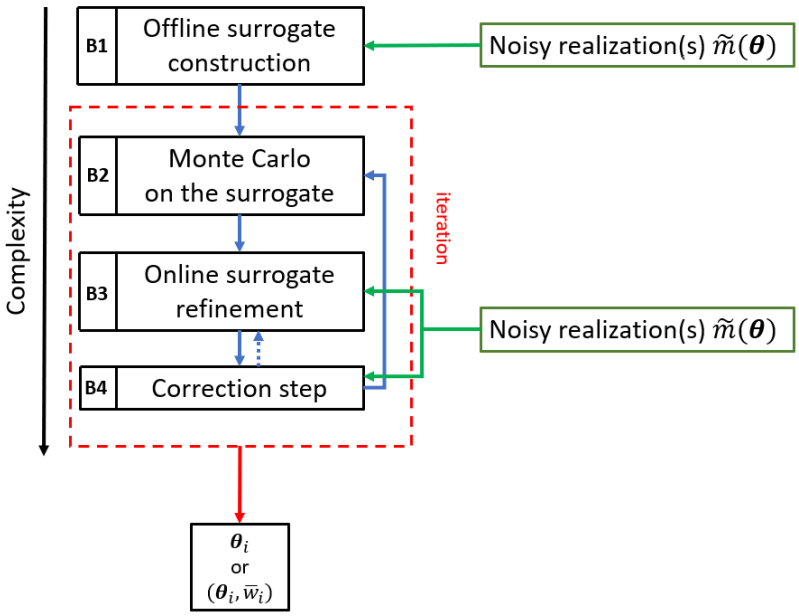


Figure 6.3: General outline of the schemes considered in the work.

Two-stage schemes (offline approximation). This scheme includes blocks 1 and 2. A *two-stage* scheme consists in running Monte Carlo algorithm on a fixed surrogate, that has

been built offline, i.e., before the start of the algorithm. This scheme is preferred when the computational budget is limited in advance, so it is all devoted to the surrogate construction. This scheme is very common in, e.g., the calibration of expensive computer codes [12, 17]. The estimators derived from this scheme are biased (w.r.t. $m(\boldsymbol{\theta})$). However, since this scheme does not imply obtaining costly realizations $\widetilde{m}(\boldsymbol{\theta})$ in the second stage, the algorithms can be run for many iterations and produce estimators with low variance. For this scheme to be worth, the decrease in variance must compensate the presence of bias.
Recent methods proposed in the literature follow this scheme. For instance, in [23], a pilot run of Monte-Carlo-within-Metropolis is carried out using unbiased estimates of the likelihood function, in order to obtain the design points and build a GP surrogate of $\log p(\boldsymbol{\theta})$. In [34], a GP regression model of $\log p(\boldsymbol{\theta})$ is built from noisy realizations by sequentially maximizing sophisticated acquisition functions, derived from Bayesian decision theory/Bayesian experimental design. In [50], they propose accelerating algorithms for doubly intractable posteriors by replacing the IS estimates (of the ratio of intractable constants) with estimates provided by a surrogate. This surrogate is built in a previous stage using GPs on the outputs of exchange algorithm runs.

Iterative refinement schemes (online approximation). This second scheme comprises Blocks 1 (optionally), 2 and 3. It considers iteratively building the surrogate along with the execution of the Monte Carlo algorithm, i.e., $\widehat{m}_t$ depends on t. In every iteration, a test is performed in order to decide if we update the surrogate (i.e., obtain a new noisy realization $\widetilde{m}(\boldsymbol{\theta})$). The surrogate refinement can be made at the end and/or beginning of the iteration (i.e., Block 3 could be placed before and/or after Block 2). This scheme is also biased, but a continual refinement of the surrogate can produce an algorithm that is asymptotically exact (in the sense of approximating $m(\boldsymbol{\theta})$) [18, 67]. See [20] for continual refinement strategies of local approximations within MCMC algorithms. Generally speaking, if the surrogate is improved infinitely often, and in a suitable way (e.g., with a space-filling strategy), the error between the surrogate $\widehat{m}_t(\boldsymbol{\theta})$ and $m(\boldsymbol{\theta})$ will approach zero. An initial surrogate $\widehat{m}_0(\boldsymbol{\theta})$ could be built offline by using some of the strategies of the methods from the previous scheme. Clearly, constantly changing the target density within a Monte Carlo algorithm difficult its analysis. Moreover, in MCMC algorithms, updating the surrogate using past states of the chain produces the loss of Markov property, so (as in the adaptive MCMC literature) one needs to carefully address this point [67, 18].
Some proposed methods that follow this scheme are [33, 18]. In [33], a GP regression model of $\log p(\boldsymbol{\theta})$ is built online by maximizing acquisition functions derived in order to decrease the uncertainty in the computation of the MH accept-reject test (i.e., using Bayesian decision theory/Bayesian experimental design). This algorithm can be considered as a two-stage procedure if we use a pilot run for the construction. In [18], a local GP or polynomial approximation is built on $\log p(\boldsymbol{\theta})$ and refined over the MCMC iterations by using space-filling heuristics.

Exact schemes (with correction step). This scheme includes blocks (optionally) 1, 2,

(optionally) 3 and 4. The main difference w.r.t. the previous schemes is the correction step. At some iterations of the method, we obtain a noisy realization $\widetilde{m}(\boldsymbol{\theta})$, in order to ensure the correctness of the algorithm, which will approximate and/or converge to $m(\boldsymbol{\theta})$. The underlying idea is to use the surrogate $\widehat{m}(\boldsymbol{\theta})$ as a (non-parametric) proposal density within a Monte Carlo method that targets $m(\boldsymbol{\theta})$. If $\widehat{m}(\boldsymbol{\theta})$ is a good approximation to $m(\boldsymbol{\theta})$, we would propose very good candidates. Working with $\widehat{m}(\boldsymbol{\theta})$ is usually cheaper than obtaining new realizations $\widetilde{m}(\boldsymbol{\theta})$. However, the fact that a new realization $\widetilde{m}(\boldsymbol{\theta})$ has to be obtained for every "correction" usually prevents significant computational savings. This scheme can be used with a fixed offline-built surrogate, an online surrogate or combination of both.
Some examples of methods leveraging surrogate models to produce efficient proposals in the literature are the following (mostly in the non-noisy context, i.e., $\widetilde{m}(\boldsymbol{\theta}) = m(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$). In the MCMC context, the delayed acceptance (DA) schemes (see next Section) are two-step MH algorithms that perform one MH iteration w.r.t. the surrogate and then compute a corrected acceptance probability for the resulting proposal in order to preserve correctness [16, 9]. Hence, DA schemes rely on approximate sampling from the surrogate via one MH step. Other works consider an standard MH algorithm where the surrogate is sampled with direct methods [44]. A rejection sampling (RS) scheme for sampling the surrogate is applied in [68], where a kriging-based surrogate is built within a delayed rejection MH [30]. In the IS context, the authors in [40] propose sampling the surrogate with IS resampling steps, and then weigh the resulting samples w.r.t. the true target.

Remark 6. *Note that the online improvement of the surrogate corresponds to the adaptation of the equivalent proposal of block B2 (see Remark 5) using not only the information of past samples, but also the history of noisy evaluations of the target.*

Table 6.3 provides some examples of methods belonging to this class and specifying the type of Monte Carlo technique in the blocks 2 and 4. Finally, Table 6.4 provides a summary of the relationship between the three main classes and the blocks 2 and 4 in Figure 6.3.

Table 6.3: Summary of specific algorithms, attending to the Blocks 2 and 4.

Exact algorithms	Block 2	Block 4
Sticky MCMC [44]	direct	MCMC
Noisy Deep IS [40]	IS	IS
Kriging AIS [8]	MCMC	IS
Delayed-acceptance MH [16, 9]	MCMC	MCMC
Kriging-based delayed rejection MH [68]	RS	MCMC

Honorable mentions. Other ways of using surrogates to improve Monte Carlo methods that do not compromise the exactness are, e.g., HMC with gradient computations based

Table 6.4: Relationship between the four main classes and the blocks (B1, B2, B3 and B4) enumerated in Figure 6.3. In parenthesis, we write the blocks that are optional to each family of methods.

Family	**Two-stage**	**Iterative refinement**	**Exact – w.r.t.** $m(\boldsymbol{\theta})$
Blocks	B1, B2	(B1), B2, B3	(B1), B2, (B3), B4

Table 6.5: Several works in the literature classified into the three presented classes.

Two-stage	**Iterative refinement**	**Exact**
[12]	[18]	[16]
[34]	[67]	[59]
[23]	[33]	[40]
[64]	[20]	[44]

on the surrogate [54]. In [27], the authors introduce extensions of the previous idea to multimodal scenarios by combining it with parallel tempering, where only the lowest temperature chain addresses the true posterior while the other chains at higher temperatures work with surrogates.

6.4. Specific instances of noisy Monte Carlo methods

In this section, we describe some specific techniques which are included in the generic scheme described in the previous section. They are Monte Carlo algorithms that were introduced mainly in the context of costly, but non-noisy, targets, but their extension to the noisy setting is straightforward. We focus on the iterative and exact families of methods, but it should be noted that the strategies for building offline surrogates (from the algorithms within the two-stage scheme) could also be used to initialize the surrogates and hence further improve these algorithms.

MH schemes on iterative surrogate. A generic MH algorithm targeting a surrogate that is refined over T iterations is given in Table 6.6. This algorithm falls within the iterative refinement scheme from the previous section. We also summarize different variants using a joint description. Indeed, at each iteration, the surrogate is updated with probability ρ_{update}, obtaining a noisy realization and including it in the set of active nodes. The different variants are obtained by designing a different probability ρ_{update} and deciding the search strategy.
Note that the updating probability could depend on many features, e.g., on the current surrogate $\rho_{\text{update}} = \rho^{(t)}_{\text{update}}(\widehat{m}_{t-1}, \psi)$ and other hyperparameters ψ. The new point to be included, $\boldsymbol{\theta}_{\text{new}}$, can be chosen by different strategies [18, 33]. As an example, in this work we will specifically consider and compare two basic algorithms. The first one corresponds

to $\rho_{\text{update}} = 1$, while the second one considers $\rho_{\text{update}} = \alpha_{\text{MH}}^{(t)}$ [67]. In both, we consider the simple choice $\boldsymbol{\theta}_{\text{new}} = \boldsymbol{\theta}'$, i.e., the new node is the proposed state at that iteration. The updating block could be also placed before the MH acceptance test and repeated until some criterion is met [18, 33].

Table 6.6: Metropolis-Hastings on surrogate with iterative refinement (MH-S)

1. **Inputs:** Initial state $\boldsymbol{\theta}_0$ and initial surrogate $\widehat{m}_0(\boldsymbol{\theta}) = \widehat{m}_0(\boldsymbol{\theta}; \mathcal{S}^{(0)})$.
2. For $t = 1, \dots, T$:
   (a) Sample $\boldsymbol{\theta}_{\text{prop}} \sim \varphi(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$.
   (b) With probability
   $$\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_{\text{prop}}) = \min\left\{1, \frac{\widehat{m}_{t-1}(\boldsymbol{\theta}_{\text{prop}})\varphi(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}_{\text{prop}})}{\widehat{m}_{t-1}(\boldsymbol{\theta}_{t-1})\varphi(\boldsymbol{\theta}_{\text{prop}}|\boldsymbol{\theta}_{t-1})}\right\}, \tag{6.6}$$
   accept $\boldsymbol{\theta}_{\text{prop}}$, i.e., set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{\text{prop}}$. Otherwise, reject $\boldsymbol{\theta}_{\text{prop}}$, i.e., set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$.
   (c) With probability ρ_{update},
      (a) Search $\boldsymbol{\theta}^\star$ and obtain realization $\widetilde{m}(\boldsymbol{\theta}^\star)$.
      (b) Update design nodes set $\mathcal{S}^{(t)} = \mathcal{S}^{(t-1)} \cup \{\boldsymbol{\theta}^\star, \widetilde{m}(\boldsymbol{\theta}^\star)\}$
3. **Outputs:** The chain $\{\boldsymbol{\theta}_t\}_{t=1}^T$ and the final surrogate $\widehat{m}_T(\boldsymbol{\theta})$.

Delayed-acceptance Metropolis-Hastings. The DA-MH algorithm is a modified MH algorithm (also called *'two-step MH'* or *'MH with early rejection'* [16, 9]) where, at each iteration, the proposed state $\boldsymbol{\theta}_{\text{prop}}$ undergoes two MH accept-reject tests. We consider here delayed-acceptance pseudo-marginal MH (DA-PM-MH), where noisy evaluations are recycled as commented above. At each iteration, the proposed state is tested first against $\widehat{m}(\boldsymbol{\theta})$ (i.e., block B2 is a MH step on the surrogate) and, upon acceptance, then against $\widetilde{m}(\boldsymbol{\theta})$ (i.e., block B4 is a noisy MH step).

The computational savings occur when $\boldsymbol{\theta}_{\text{prop}}$ is rejected in the first test, since it avoids performing the second MH test and computing the costly noisy realization $\widetilde{m}(\boldsymbol{\theta}_{\text{prop}})$. In this work, we consider a general version DA-PM-MH (also called surrogate transition method [39]) that allows for multiple iterations w.r.t. $\widehat{m}$ in the first step. The details are given in Table 6.7. The standard DA-PM-MH algorithm is recovered setting $T_{\text{surr}} = 1$. The standard DA-PM-MH has always a lower acceptance than vanilla PM-MH [9], but can provide better performance. However, for $T_{\text{surr}} \geq 1$, the acceptance probability can be higher than in the standard MH.
Indeed, this general form of the DA-PM-MH algorithm makes it clear that first step aims at obtaining a good candidate $\boldsymbol{\xi}_{T_{\text{surr}}}$ by sampling (via MCMC) from a proposal density $\widehat{m}$ built by a (usually non-parametric) surrogate model. The candidate sample $\boldsymbol{\xi}_{T_{\text{surr}}}$ is then employed in a MH test w.r.t. $\widetilde{m}$. It is important to note that, if all tests in the secondary chain got rejected, then $\boldsymbol{\theta}_{\text{prop}} = \boldsymbol{\xi}_{T_{\text{surr}}} = \boldsymbol{\theta}_{t-1}$, so the MH test of the main chain is trivially

accepted without needing to obtain a new noisy realization, i.e., the chain remains at $\boldsymbol{\theta}_{t-1}$. We can interpret the DA-MH as a two-step algorithm where, in the first step, samples approximately distributed as the surrogate are generated. Thus, other algorithms such as sticky MCMC [44] can be considered as 'ideal' version of DA-MH, since the samples are drawn directly from the surrogate (i.e. the acceptance probability in the first step is always one).

Table 6.7: DA-PM-MH algorithm

1. **Inputs:** Initial state $\boldsymbol{\theta}_0$, initial realization $\widetilde{m}(\boldsymbol{\theta}_0)$, surrogate $\widehat{m}_0(\boldsymbol{\theta}; \mathcal{S}^{(0)})$, and number of 'inner' iterations T_{surr}.
2. For $t = 1, \ldots, T$:
   (a) Starting from $\boldsymbol{\theta}_{t-1}$, run T_{surr} iterations of MH with respect to $\widehat{m}(\boldsymbol{\theta})$. That is, set $\boldsymbol{\xi}_0 = \boldsymbol{\theta}_{t-1}$ and do for $k = 1, \ldots, T_{\text{surr}}$:
      (i) Sample $\boldsymbol{\xi}' \sim \varphi(\boldsymbol{\theta}|\boldsymbol{\xi}_{k-1})$
      (ii) With probability
      $$\alpha_1(\boldsymbol{\xi}_{k-1}, \boldsymbol{\xi}') = \min\left\{1, \frac{\widehat{m}_{t-1}(\boldsymbol{\xi}')\varphi(\boldsymbol{\xi}_{k-1}|\boldsymbol{\xi}')}{\widehat{m}_{t-1}(\boldsymbol{\xi}_{k-1})\varphi(\boldsymbol{\xi}'|\boldsymbol{\xi}_{k-1})}\right\},$$
      accept $\boldsymbol{\xi}'$, i.e., set $\boldsymbol{\xi}_k = \boldsymbol{\xi}'$. Otherwise, reject $\boldsymbol{\xi}'$, i.e., set $\boldsymbol{\xi}_k = \boldsymbol{\xi}_{k-1}$.
   (b) Set $\boldsymbol{\theta}_{\text{prop}} = \boldsymbol{\xi}_{T_{\text{surr}}}$, and accept it with probability
   $$\alpha_1(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_{\text{prop}}) = \min\left\{1, \frac{\widetilde{m}(\boldsymbol{\theta}_{\text{prop}})\widehat{m}_{t-1}(\boldsymbol{\theta}_{t-1})}{\widetilde{m}(\boldsymbol{\theta}_{t-1})\widehat{m}_{t-1}(\boldsymbol{\theta}_{\text{prop}})}\right\},$$
   i.e., set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{\text{prop}}$. Otherwise, reject $\boldsymbol{\theta}_{\text{prop}}$, i.e., set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$.
   (c) With probability ρ_{update},
      1. Search $\boldsymbol{\theta}^\star$ and obtain realization $\widetilde{m}(\boldsymbol{\theta}^\star)$.
      2. Update design nodes set $\mathcal{S}^{(t)} = \mathcal{S}^{(t-1)} \cup \{\boldsymbol{\theta}^\star, \widetilde{m}(\boldsymbol{\theta}^\star)\}$
3. **Outputs:** The chain $\{\boldsymbol{\theta}_t\}_{t=1}^T$.

Noisy Deep Importance Sampling (N-DIS). The Deep Importance Sampling (DIS) is an adaptive IS scheme introduced in [40], which uses a non-parametric surrogate as its proposal density. It can be seen as a multivariate extension of the technique in [44]. Here, we consider a noisy version of DIS, which is described in Table 6.8. Again the underlying idea is to use the surrogate $\widehat{m}(\boldsymbol{\theta})$ as proposal density. For sampling from $\widehat{m}(\boldsymbol{\theta})$, N-DIS employs a Sampling Importance Resampling (SIR) approach [57], using an auxiliary/parametric proposal, $q(\boldsymbol{\theta})$ (i.e., block B1 is a SIR scheme). More specifically, a set $\{\mathbf{y}\}_{\ell=1}^L$ is sampled from $q(\boldsymbol{\theta})$, with $L \gg 1$, and weighted according to $\widehat{m}$. Then, N resampling steps ($N \ll L$) are performed to obtain $\{\boldsymbol{\theta}_i\}_{i=1}^N$, that are approximately distributed as $\widehat{m}(\boldsymbol{\theta})$ [57]. These samples are finally weighted considering the corresponding realizations $\widetilde{m}(\boldsymbol{\theta}_i)$ (i.e., block B4 is a IS iteration). Thus, N-DIS is as a two-stage IS scheme,

where the inner IS stage is employed to draw from the surrogate $\widehat{m}$. Furthermore, N-DIS is an iterative algorithm where the previous steps are repeated and the set $\{\boldsymbol{\theta}_i\}_{i=1}^N$ is used to refine the surrogate at each iteration. Hence, compared to a standard IS scheme, N-DIS improves the performance by using a non-parametric surrogate proposal density $\widehat{m}(\boldsymbol{\theta})$ that gets closer and closer to $m(\boldsymbol{\theta})$. Moreover, N-DIS could be interpreted as an IS version equivalent to the DA-MH algorithm. Note that, N-DIS uses deterministic mixture IS weights in Eq. (6.7) which provide more stability in the results [19].

Table 6.8: N-DIS algorithm with noisy realizations

1. **Inputs:** Proposal distribution $q(\boldsymbol{\theta})$ and initial surrogate $\widehat{m}_0(\boldsymbol{\theta}) = \widehat{m}_0(\boldsymbol{\theta}; \mathcal{S}^{(0)})$.
2. For $t = 1, \ldots, T$:
   (a) Sample $\boldsymbol{\xi}_{t,\ell} \sim q(\boldsymbol{\theta})$, $\ell = 1, \ldots, L$
   (b) Compute $\gamma_{t,\ell} = \dfrac{\widehat{m}_{t-1}(\boldsymbol{\xi}_{t,\ell})}{q(\boldsymbol{\xi}_{t,\ell})}$, $\ell = 1, \ldots, L$
   (c) Resample $\boldsymbol{\theta}_{t,n} \sim \{\boldsymbol{\xi}_{t,\ell}\}_{\ell=1}^L$, $n = 1, \ldots, N$, with probabilities proportional to $\{\gamma_{t,\ell}\}_{\ell=1}^L$ (with $N \ll L$)
   (d) Obtain noisy realizations and compute ($n = 1, \ldots, N$)
   $$w_{t,n} = \frac{\widetilde{m}(\boldsymbol{\theta}_{t,n})}{\frac{1}{t}\sum_{\tau=0}^{t-1} \widehat{m}_\tau(\boldsymbol{\theta}_{t,n})}; \tag{6.7}$$
   (e) Update design nodes set $\mathcal{S}^{(t)} = \mathcal{S}^{(t-1)} \cup \{(\boldsymbol{\theta}_{t,n}, \widetilde{m}(\boldsymbol{\theta}_{t,n}))\}_{n=1}^N$.
3. Compute normalized weights: $\bar{w}_n = \frac{w_n}{\sum_{j=1}^N w_j}$, $j = 1, \ldots, N$.
4. **Outputs:** the weighted samples $\{\boldsymbol{\theta}_n, \bar{w}_n\}_{n=1}^N$ and the final surrogate $\widehat{m}_t(\boldsymbol{\theta})$.

6.5. Application scenarios

A brief description of practical scenarios where we must handle noisy and costly target evaluations is provided below. Namely, all the settings given below can be encompassed in the general framework described above.

Pseudo-Marginal approach: Here, the unnormalized density can be expressed as a marginal distribution, i.e., $p(\boldsymbol{\theta}) = \int_{\mathcal{V}} p(\boldsymbol{\theta}, \mathbf{v}) d\mathbf{v}$ where $\mathbf{v}$ is an auxiliary variable. Hence, $p(\boldsymbol{\theta})$ cannot be computed in closed-form. When the aim is to run a MH algorithm on $p(\boldsymbol{\theta})$, rather than on the joint $p(\boldsymbol{\theta}, \mathbf{v})$, the evaluation of $p(\boldsymbol{\theta})$ at each $\boldsymbol{\theta}$ can be estimated *noisily* by using IS [5, 4].

ABC, likelihoods-free. In the likelihood-free (and/or synthetic likelihood) inference setting, it is assumed that the likelihood function is unknown or we cannot evaluate it, but we are able to generate independent data from it. In this scenario, substituting the intractable likelihood with an approximate likelihood is one possibility. This approximation is in turn approximated pointwise with Monte Carlo using pseudo-data sets [43, 53]. See Section 6.5.1 below for more details.

Doubly intractable posteriors. When only a part of the likelihood can be evaluated and another piece of the likelihood is unknown (typically a partition function $Z(\boldsymbol{\theta})$), we are in doubly intractable posterior setting. Note that differently from the ABC case, here some part of the likelihood is available. In this case, the unknown part of the likelihood must be estimated, so that the evaluation of the complete likelihood will be noisy.

Use of mini-batches (Big Data). The evaluation of the likelihood function can be prohibitively expensive when there are huge amounts of data. In this context, a *subsampling* strategy consists in computing the log-likelihood function using a random subset of data points, hence forming an unbiased estimator of the complete log-likelihood [10].

Reinforcement learning (RL). Direct policy search is an important branch of reinforcement learning, particularly in robotics [21, 15]. In this context, $\boldsymbol{\theta}$ is the parametrization of the policy of some agent, and $p(\boldsymbol{\theta})$ represents an expected return (i.e., a payoff function) for that policy. The expected return is approximated by the empirical return over an episode, i.e., the agent is run for a number of time steps and accumulates a payoff. More details are given in Section 6.5.2.

Other application scenarios. The topic of inference in noisy and costly settings is also of interest in the inverse problem literature, such as in the calibration of expensive computer codes [24, 17, 12]. Noisy likelihood evaluations are also considered for building surrogates, and then use them in order to obtain a variational approximations to the posterior [1].

6.5.1. Likelihood-free context

The Likelihood-free framework in Bayesian inference presents some peculiarities which deserve a specific discussion. We start with a brief description of a generalized approximate Bayesian computation (ABC) scheme in the same fashion of [66, 52]. Given some

vector of data $\mathbf{y}_{\texttt{true}} \in \mathbb{R}^{D_Y}$, in several applications, sampling from a posterior distribution $p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y}_{\texttt{true}}) \propto \ell(\mathbf{y}_{\texttt{true}}|\boldsymbol{\theta})g(\boldsymbol{\theta})$ is required, where $\ell(\mathbf{y}_{\texttt{true}}|\boldsymbol{\theta})$ represents a likelihood function and $g(\boldsymbol{\theta})$ a prior density. In some context, the pointwise evaluation of $\ell(\mathbf{y}_{\texttt{true}}|\boldsymbol{\theta})$ is not possible, but we can generate artificial data, $\mathbf{y}' \sim \ell(\mathbf{y}|\boldsymbol{\theta})$. Hence, we could draw samples in an extended space, $[\boldsymbol{\theta}', \mathbf{y}']$, from the joint pdf $q(\boldsymbol{\theta}, \mathbf{y}) = \ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})$, drawing first $\boldsymbol{\theta}' \sim g(\boldsymbol{\theta})$ and then $\mathbf{y}' \sim \ell(\mathbf{y}|\boldsymbol{\theta})$.
The idea behind several ABC algorithms is the following. Let us consider the following extended target pdf in the extended space $[\boldsymbol{\theta}, \mathbf{y}]$,

$$p_e(\boldsymbol{\theta}, \mathbf{y}|\mathbf{y}_{\texttt{true}}, \epsilon) \propto h(\mathbf{y}_{\texttt{true}}|\mathbf{y}, \boldsymbol{\theta}, \epsilon)\ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta}),$$

where $h(\mathbf{y}_{\texttt{true}}|\mathbf{y}, \boldsymbol{\theta}, \epsilon) \geq 0$ is a *surrogate extended likelihood* and $\epsilon > 0$ is a positive parameter, chosen by the user. In many ABC approaches, different authors consider a simplified version where

$$h(\mathbf{y}_{\texttt{true}}|\mathbf{y}, \boldsymbol{\theta}, \epsilon) = h(\mathbf{y}_{\texttt{true}}|\mathbf{y}, \epsilon),$$

for instance, $h(\mathbf{y}_{\texttt{true}}|\mathbf{y}, \epsilon) \propto \exp\left(-\frac{\|\mathbf{y}_{\texttt{true}}-\mathbf{y}\|^2}{2\epsilon^2}\right)$. Hence, we can simplify the previous expression as $p_e(\boldsymbol{\theta}, \mathbf{y}|\mathbf{y}_{\texttt{true}}) \propto h(\mathbf{y}_{\texttt{true}}|\mathbf{y}, \epsilon)\ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})$. The simplest choice, as in the rejection-ABC scheme, is

$$\begin{cases} h(\mathbf{y}_{\texttt{true}}|\mathbf{y}, \epsilon) \propto 1 & \text{if } \|\mathbf{y}_{\texttt{true}} - \mathbf{y}\| < \epsilon, \\ h(\mathbf{y}_{\texttt{true}}|\mathbf{y}, \epsilon) = 0 & \text{if } \|\mathbf{y}_{\texttt{true}} - \mathbf{y}\| \geq \epsilon. \end{cases} \tag{6.8}$$

Therefore, the ABC target density is

$$m_{\text{ABC}}(\boldsymbol{\theta}|\mathbf{y}_{\texttt{true}}, \epsilon) = \int_{\mathbb{R}^{D_Y}} p_e(\boldsymbol{\theta}, \mathbf{y}|\mathbf{y}_{\texttt{true}}, \epsilon)d\mathbf{y} \propto \int_{\mathbb{R}^{D_Y}} h(\mathbf{y}_{\texttt{true}}|\mathbf{y}, \epsilon)\ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})d\mathbf{y}. \tag{6.9}$$

The function $h(\mathbf{y}_{\texttt{true}}|\mathbf{y}, \epsilon)$ must be chosen such that $m_{\text{ABC}}(\boldsymbol{\theta}|\mathbf{y}_{\texttt{true}}, \epsilon)$ converges to $p(\boldsymbol{\theta}|\mathbf{y}_{\texttt{true}})$ as $\epsilon \to 0$. Several computational algorithms designed for the ABC context are based on the following *noisy naive Monte Carlo* scheme in the extended space with target pdf $m_{ABC}(\boldsymbol{\theta}|\mathbf{y}_{\texttt{true}}, \epsilon)$ in Eq. (6.9), and proposal density $q(\boldsymbol{\theta}, \mathbf{y}) = \ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})$:

- For $t = 1, ..., T$:
  1. Draw $\boldsymbol{\theta}_t \sim g(\boldsymbol{\theta})$,
  2. Draw N artificial data, $\mathbf{y}_t^{(1)}, \ldots, \mathbf{y}_t^{(N)} \sim \ell(\mathbf{y}|\boldsymbol{\theta}_t)$.[11]
  3. Assign to $\boldsymbol{\theta}_t$, the noisy evaluation

$$\widetilde{m}_\epsilon(\boldsymbol{\theta}_t) = \frac{1}{N}\sum_{n=1}^{N} h(\mathbf{y}_{\texttt{true}}|\mathbf{y}_t^{(n)}, \epsilon). \tag{6.10}$$

- Return $\{\boldsymbol{\theta}_t, \widetilde{m}(\boldsymbol{\theta}_t)\}$.

[11] Note that $\{\boldsymbol{\theta}_t, \mathbf{y}_t^{(n)}\} \sim q(\boldsymbol{\theta}, \mathbf{y})$ for all n. See the generalized chain rule in [45].

Thus, the pairs $\{\boldsymbol{\theta}_t, \widetilde{m}_\epsilon(\boldsymbol{\theta}_t)\}$ can be used for performing inference on $m_{\text{ABC}}(\boldsymbol{\theta}|\mathbf{y}_{\text{true}}, \epsilon)$. Indeed, by standard Monte Carlo arguments, $\widetilde{m}_\epsilon(\boldsymbol{\theta}) \approx \int_{\mathbb{R}^{D_Y}} h(\mathbf{y}_{\texttt{true}}|\mathbf{y}, \epsilon)\ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})d\mathbf{y}$. Increasing N, we reduce the variance of $\widetilde{m}_\epsilon(\boldsymbol{\theta})$, becoming closer and closer to $m_{ABC}(\boldsymbol{\theta}|\mathbf{y}_{\texttt{true}}, \epsilon)$. Decreasing $\epsilon \to 0$, we reduce the bias between $m_{ABC}(\boldsymbol{\theta}|\mathbf{y}_{\texttt{true}}, \epsilon)$ and $p(\boldsymbol{\theta}|\mathbf{y}_{\texttt{true}})$. Instead of sampling $\boldsymbol{\theta}_t$ from $g(\boldsymbol{\theta})$, we can use a generic proposal $q(\boldsymbol{\theta})$ (i.e., $q(\mathbf{y}, \boldsymbol{\theta}) = \ell(\mathbf{y}|\boldsymbol{\theta})q(\boldsymbol{\theta})$) and we obtain

$$\widetilde{m}_\epsilon(\boldsymbol{\theta}_t) = \left[\frac{1}{N}\sum_{n=1}^{N} h(\mathbf{y}_{\texttt{true}}|\mathbf{y}_t^{(n)}, \epsilon)\right]\frac{g(\boldsymbol{\theta}_t)}{q(\boldsymbol{\theta}_t)}, \qquad \theta_t \sim q(\boldsymbol{\theta}). \tag{6.11}$$

Remark 7. *Clearly, for a fixed computational cost, there exists a trade-off between exploration and accuracy, i.e., between T and N. For a related discussion, see [22, 38].*

Since simulating N datasets for each $\boldsymbol{\theta}$ can be costly, it has been proposed to use surrogates in order to accelerate the ABC algorithms. For instance, we can build a surrogate $\widehat{m}(\boldsymbol{\theta})$ considering the pairs $\{\boldsymbol{\theta}_t, \widetilde{m}(\boldsymbol{\theta}_t)\}$ or some related evaluations. In [66], a two-stage approach is used, where a GP surrogate of $\log m_{\text{ABC}}$ is built offline, and then a random-walk MH algorithm is applied on this surrogate An iterative refinement scheme using simulations $(\boldsymbol{\theta}_t, \mathbf{y}_t^{(n)})$ is considered in [47]. Finally, the work by [29] combines Bayesian optimization with ABC in a two-stage scheme to build a surrogate of the discrepancy function $\Delta_{\boldsymbol{\theta}}$ which measures the difference between $\mathbf{y}_{\text{true}}$ and $\mathbf{y}_{\boldsymbol{\theta}}$, the data generated with parameter $\boldsymbol{\theta}$.

Remark 8. *In the ABC context, we identify two surrogate functions: an internal surrogate $h(\mathbf{y}_{\texttt{true}}|\mathbf{y}, \boldsymbol{\theta}, \epsilon)$ (that, generally, could also depends on $\boldsymbol{\theta}$ as in the synthetic likelihood approach [53]) and the external surrogate $\widehat{m}(\boldsymbol{\theta})$, for accelerating the algorithm.*

6.5.2. Application to Reinforcement Learning

Reinforcement learning (RL), which has many connections with control theory [28, 60], is a popular and fast-growing area of machine learning. An agent interacts with an environment by taking an action and, as a result of this action, it receives a state/observation and a reward. This occurs at each time step. One interaction/step is summarized as a state-action-reward triplet, (s_t, a_t, r_t), where t denotes the time index. Therefore, an episode consists of T steps over the environment (e.g., playing a game, if the environment represents a game, or otherwise interacting with the environment – such as in robotics)

$$\boldsymbol{\tau} = \{s_0, (s_1, a_1, r_1), (s_2, a_2, r_2), \ldots, (s_T, a_T, r_T)\} = \{s_{0:T}, a_{1:T}, r_{1:T}\}. \tag{6.12}$$

The dynamics of the environment can be represented as follows, in the case of Markovian processes. For $t = 1, 2, \ldots, T$:

$$\begin{cases} a_t \sim \pi_{\boldsymbol{\theta}}(a|s_{t-1}), \\ s_t \sim p_{\text{env}}(s|s_{t-1}, a_t), \\ r_t \sim r_{\text{env}}(r|s_t, a_t, s_{t-1}), \end{cases} \tag{6.13}$$

where the reward function r_{env} and the transition function p_{env} are determined by the application/environment. The policy $\pi_{\boldsymbol{\theta}}(\cdot)$ determines which action the agent takes. Deterministic rules can be also employed for deciding a_t and receiving a reward r_t. The payoff (i.e., accumulated reward, known as the return, or gain) for each episode is

$$R(\boldsymbol{\theta};\boldsymbol{\tau}) = \sum_{t=1}^{T} r_t. \tag{6.14}$$

In certain settings, one can control the length T of the episode $\boldsymbol{\tau}$. The goal is to find an optimal policy (i.e., optimal $\boldsymbol{\theta}$) that maximizes the expected cumulative reward. There are a plethora of approaches to reinforcement learning, many falling under the category of so-called value-based methods (see [60] for an introduction and overview). Here, however, we focus specifically on the area of direct policy search, which is particularly apt for applications with continuous and small-but-complex action spaces such as robotics [21], and possibly non-Markovian settings (we refer to p_{env}). More specifically, we focus on model-free policy search, i.e., learning the policy based on sampling trajectories; we do not attempt to recover p_{env} or r_{env}. In this sense also, we are close to the large area of stochastic optimization [51]. We are interested in studying the following function in the parameter space,

$$p(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\tau}}[R(\boldsymbol{\theta};\boldsymbol{\tau})] = \int_{\mathcal{T}} R(\boldsymbol{\theta};\boldsymbol{\tau}) p(\boldsymbol{\tau}|\boldsymbol{\theta}) d\boldsymbol{\tau}, \tag{6.15}$$

where $R(\cdot)$ from (6.14), and $\boldsymbol{\tau} \sim p(\boldsymbol{\tau}|\boldsymbol{\theta})$ is generated following the model in Eq. (6.13), i.e.,

$$\begin{aligned} p(\boldsymbol{\tau}|\boldsymbol{\theta}) &= p(s_{0:T}, a_{0:T}, r_{1:T}|\boldsymbol{\theta}), \\ &= p_0(s_0) \prod_{t=1}^{T} r_{\text{env}}(r_t|s_t, a_t, s_{t-1}) p_{\text{env}}(s_t|s_{t-1}, a_{t-1}) \pi_{\boldsymbol{\theta}}(a_{t-1}|s_{t-1}). \end{aligned} \tag{6.16}$$

In a model-free direct search, we are not able to evaluate the distribution $p(\boldsymbol{\tau}|\boldsymbol{\theta})$, but we can draw from it by "playing the game". Namely, we can estimate $p(\boldsymbol{\theta})$ by using sampled episodes. Given N episodes $\boldsymbol{\tau}_i \sim p(\boldsymbol{\tau}|\boldsymbol{\theta})$ $(i = 1, \ldots, N)$ generated according to $p(\boldsymbol{\tau}|\boldsymbol{\theta})$ with fixed $\boldsymbol{\theta}$ (and fixing T), we can obtain the Monte Carlo estimation of the expected return

$$\widetilde{m}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} R(\boldsymbol{\theta};\boldsymbol{\tau}_i), \qquad \boldsymbol{\tau}_i \sim p(\boldsymbol{\tau}|\boldsymbol{\theta}), \tag{6.17}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} r_t^{(i)}. \tag{6.18}$$

In this case, we have $m(\boldsymbol{\theta}) = \mathbb{E}[\widetilde{m}(\boldsymbol{\theta})] = p(\boldsymbol{\theta})$. The variance

$$s^2(\boldsymbol{\theta}) = \mathtt{var}\left[\widetilde{m}(\boldsymbol{\theta})\right] = \frac{1}{N} \text{var}\left[R(\boldsymbol{\theta};\boldsymbol{\tau}_i)\right]. \tag{6.19}$$

The term $\text{var}[R(\boldsymbol{\theta},\boldsymbol{\tau})]$ can have different forms depending on multiple aspects. The magnitude of the noise is reduced by averaging multiple episodes since the variance $s^2(\boldsymbol{\theta})$ decreases at rate $\frac{1}{N}$.

Remark 9. *As in the ABC setting, there is clearly a trade-off between precision in the evaluation of the target function and overall computational cost (which increases as N grows). This trade-off has been studied in the context of MCMC and IS [22, 38].*

Note that the distribution $\widetilde{m}(\boldsymbol{\theta})$ also depends of the length T of the episode. More specifically, the variance of the random variable $\widetilde{m}(\boldsymbol{\theta})$ decreases with T. If the process is ergodic, averaging over very long periods is equivalent to repeating the process multiple times. The noise can therefore be reduced by both prolonged simulation or repeated sampling at the expense of a higher computational cost per function evaluation.

6.6. Numerical experiments

In this section, we compare different algorithms discussed in Section 4. It is important to remark that all the techniques are always compared with the same number of evaluations (denoted as E) of the noisy target pdf. Moreover, a k-nearest neighbor (kNN) regression is applied in order to construct the surrogate function. Recall that the baseline PM-MH algorithm is not using a surrogate model (see Table 6.1).
In the first experiment, the target is a two-dimensional banana-shaped density which is non-linear benchmark in the literature [19], perturbed with two different noises: one is an unbiased noise, and with the other noisy the target distribution becomes a heavy-tailed banana pdf. The second experiment considers a multimodal target density. Finally, we apply the algorithms in a benchmark RL problem consisting on balancing two poles attached to a cart.

6.6.1. Non-linear banana density

We consider a banana-shaped target pdf,

$$p(\boldsymbol{\theta}) \propto \exp\left(-\frac{(\eta_1 - B\theta_1 - \theta_2^2)^2}{2\eta_0^2} - \frac{\theta_1^2}{2\eta_1^2} - \frac{\theta_2^2}{2\eta_2^2}\right), \tag{6.20}$$

with $B = 4$, $\eta_0 = 4$ and $\eta_i = 3.5$ for $i = 1, \ldots, 2$, where $\Theta = [-10, 10] \times [-10, 10]$, i.e., bounded domain. The goal is to compare the performance of the different algorithms against a vanilla PM-MH algorithm for two different noises. Specifically, we compare

- **(1)** DA-PM-MH with $T_{\text{surr}} = 1$,
- **(2)** DA-PM-MH with $T_{\text{surr}} = 5$,
- **(3)** MH-S with $\rho_{\text{update}} = 1$,
- **(4)** MH-S with $\rho_{\text{update}} = \alpha_{\text{MH}}$.

The baseline corresponds to a PM-MH algorithm with 5000 iterations. We consider the same proposal $\varphi(\boldsymbol{\theta}|\boldsymbol{\theta}') = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}', 3^2\boldsymbol{I}_2)$ for all the methods (including the baseline). The surrogate is built with k-nearest neighbor (kNN) regression using $K \in \{1, 10, 100\}$ neighbors. For all methods, the surrogate is initialized as a uniform distribution and updated from there on using the incoming realizations $\widetilde{m}(\boldsymbol{\theta})$. Note that MH-S with $\rho = 1$ is equivalent to PM-MH when $K = 1$. We include it in that case for the sake of completeness. We set $E = 5000$ as the budget of noisy target evaluations.

In addition, we have applied IS schemes for the two noises. Specifically, we compare standard (noisy) IS against N-DIS, using again the nearest neighbor surrogate. For the standard noisy IS, we use a uniform proposal in $\mathcal{X}$. For N-DIS, we test $T = 5, N = 1000$ and $T = 10, N = 500$, so that the total number of evaluations is $E = NT = 5000$.

Unbiased banana. First, we consider the noise $\widetilde{m}(\boldsymbol{\theta}) = \epsilon p(\boldsymbol{\theta})$ with $\epsilon \sim \text{Exp}(1)$. In this case, the expected target is $p(\boldsymbol{\theta})$. We consider the estimation of the mean and the diagonal of the covariance matrix, whose ground truths are $\boldsymbol{\mu} = [-0.48, 0]$ and $\text{diag}(\boldsymbol{\Sigma}) = [1.38, 8.90]$. We show the results in Figures 6.4 and 6.6.

Heavy-tailed banana. Then, we consider the noise $\widetilde{m}(\boldsymbol{\theta}) = \max(0, p(\boldsymbol{\theta}) + \epsilon)$ with $\epsilon \sim \mathcal{N}(0, 0.01^2)$. For this choice, we have $m(\boldsymbol{\theta}) \neq p(\boldsymbol{\theta})$, so we have to evaluate the performance in the estimation of the new moments, i.e., this noise changes the density that the methods target, whose ground truths are $\widetilde{\boldsymbol{\mu}} = [-0.38, 0]$ and $\text{diag}(\widetilde{\boldsymbol{\Sigma}}) = [6.74, 12.84]$. The resulting density $m(\boldsymbol{\theta})$ has constant tails since this noise introduce bias in the low probability regions (as in Figure 6.1). We show the results in Figures 6.5 and 6.7.

Dependence on the surrogate

The use of surrogate improves the performance, but can be detrimental as well. This duality accounts for the differences in performance between estimating $\boldsymbol{\mu}$ (upper rows of Figures 6.4 and 6.5) and estimating $\text{diag}(\boldsymbol{\Sigma})$ (lower rows of Figures 6.4 and 6.5).

Benefits of using surrogates. For both noises, the considered algorithms perform better than the baseline in the estimation of $\boldsymbol{\mu}$ for all K, something that it is related to properly visiting the regions of high probability. In this sense, it shows that using surrogates within MCMC algorithms help in discovering high-probability regions. In IS, the use of surrogates also improves the performance in the estimation of the mean, as it can be seen in Figure 6.9(a) and Figure 6.7(a).

Pathological constructions. Both choices of noise produce noisy realizations $\widetilde{m}(\boldsymbol{\theta})$ that are skewed towards 0, specially in the low-probability regions. A surrogate built with such evaluations may difficult the exploration of the tails of the distribution. This can be seen at the error in estimating the variance in Figure 6.4(d) and Figure 6.5(d), where the considered methods perform worse than the baseline. Although the DA-PM-MH algorithms (with $T_{\text{surr}} = 1$ and $T_{\text{surr}} = 5$) are "exact", they fail at estimating the variance since the surrogate does not fulfill the minimum requirements. In fact, a 'bad' surrogate is preventing the chain to explore the regions properly. Increasing K makes the surrogate smoother and hence should improve the variance estimation. This is confirmed in Figure 6.4(e)-(f)

and Figure 6.5(e)-(f), where the DA-PM-MH algorithms perform better than the baseline. The MH-S algorithms present a trade-off between performance and exactness/bias as we increase K, that we comment below.

Bias in iterative refinement algorithms

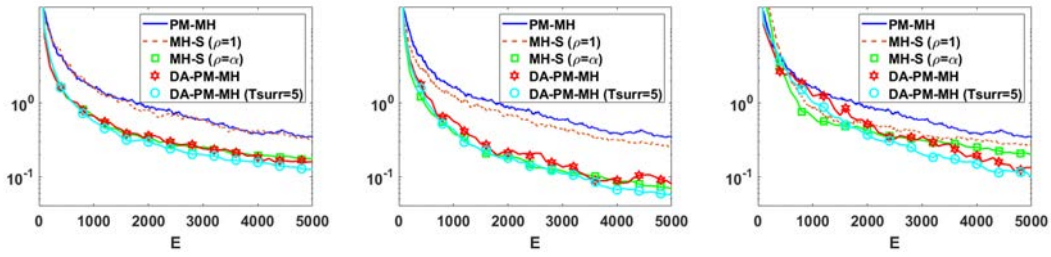
Since these algorithms target the surrogate, the choice of K affects the performance. In Figures 6.4(a)-(c), we see the algorithms MH-S with $\rho = 1$ and $\rho = \alpha$ beat the baseline in the estimation of $\boldsymbol{\mu}$. However, in Figures 6.4(d)-(f) the situation is the opposite, performing worse than the baseline in the estimation of diag($\boldsymbol{\Sigma}$) for the K considered. As we commented above, the exponential distribution with $\lambda = 1$ concentrates around 0, hence this noise tends to give noisy realizations that underestimate the true density. In low-probability regions and when $K = 1$, this phenomenon amplifies since realizations with very low value difficult that their neighborhood gets properly explored. This is why MH-S is able to estimate $\boldsymbol{\mu}$ with $K = 1$ (i.e. the high-probability region is properly visited), but fails at estimating diag($\boldsymbol{\Sigma}$).
We increase K in order to reduce this problem. However, attending to Figures 6.4(e)-(f) for $K = 10$ and $K = 100$, both MH-S still perform poorly in the estimation of diag($\boldsymbol{\Sigma}$). Now, this is because the surrogate has huge bias (since, for fixed number of nodes, as we consider more neighbors, the surrogate becomes a flattened version of $p(\boldsymbol{\theta})$). In other words, regarding the choice of K for the MH-S, the increase in performance is traded off with exactness. Note that this bias is detected when estimating the variance, since this biased surrogate has $\boldsymbol{\mu}$ almost unaltered.
Regarding the second type of noise in Figure 6.5, the conclusions are similar. In Figure 6.5(d), we see that estimation of the variance is even worse with this second noise, since the target has now constant tails which are not captured by the surrogate with $K = 1$. However, MH-S algorithms perform better (w.r.t. the previous noise) in the estimation of the variance for $K = 10$. This is probably due to the surrogate having a low bias w.r.t. the true target $m(\boldsymbol{\theta})$, which is broader than in the previous noise.

6.6.2. Bimodal target density

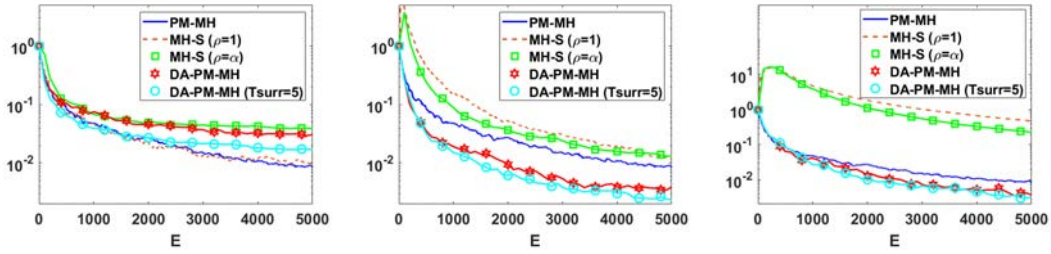
Now, we consider the density

$$p(\boldsymbol{\theta}) = \frac{1}{2}\mathcal{N}(\boldsymbol{\theta}|[10,0]^\top, 3^2\boldsymbol{I}_2) + \frac{1}{2}\mathcal{N}(\boldsymbol{\theta}|[-10,0]^\top, 3^2\boldsymbol{I}_2),$$

where $\Theta = [-20,20] \times [-20,20]$, i.e., bounded domain. We consider the noise $\widetilde{m}(\boldsymbol{\theta}) = \epsilon p(\boldsymbol{\theta})$ with $\epsilon \sim Exp(1)$. As in the previous experiment, we compare the algorithms in the estimation of the mean $\boldsymbol{\mu} = [0,0]^\top$ and the diagonal of the covariance matrix diag($\boldsymbol{\Sigma}$) = $[108.87, 9]^\top$. For the MCMC algorithms, this time we consider a proposal, $\varphi(\boldsymbol{\theta}|\boldsymbol{\theta}') = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}', 2^2\boldsymbol{I}_2)$, intentionally chosen so that the mixing can be slow for some initializations. We set $E = 5000$ as the budget of noisy evaluations. Results are shown in Figure 6.8. The

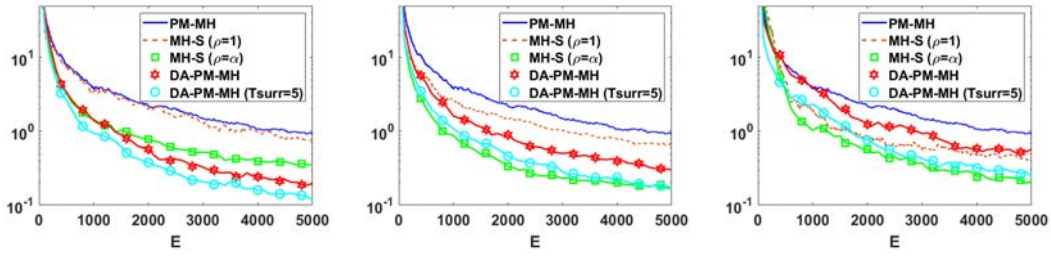


(a) $K = 1$ (b) $K = 10$ (c) $K = 100$

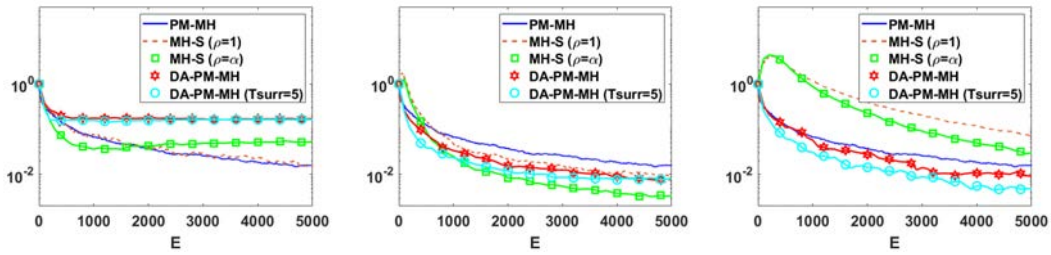


(d) $K = 1$ (e) $K = 10$ (f) $K = 100$

Figure 6.4: Relative median squared error in estimation of the mean (upper row) and variance (lower row) of the banana pdf with multiplicative exponential noise.



(a) $K = 1$ (b) $K = 1$ (c) $K = 1$

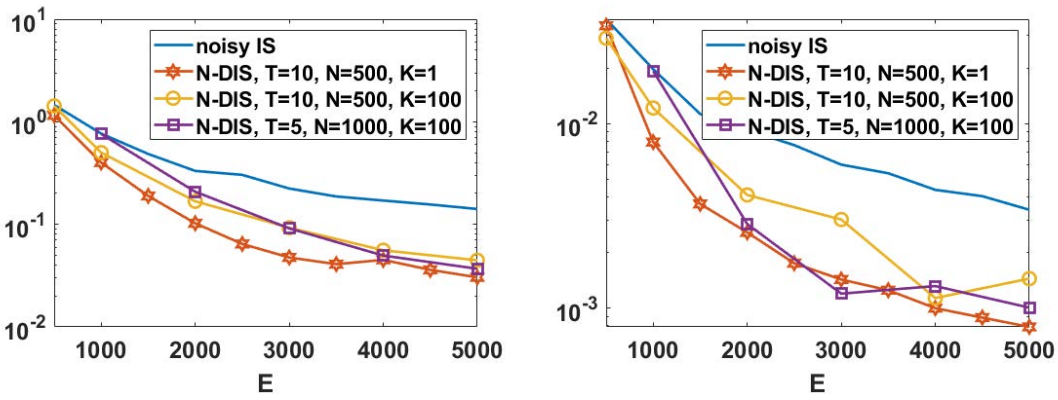


(d) $K = 1$ (e) $K = 1$ (f) $K = 1$

Figure 6.5: Relative median squared error in estimation of the mean (upper row) and variance (lower row) of the banana pdf perturbed as $\widetilde{f}(\boldsymbol{\theta}) = \max(0, p(\boldsymbol{\theta}) + \epsilon)$, $\epsilon \sim \mathcal{N}(0, 0.01)$.

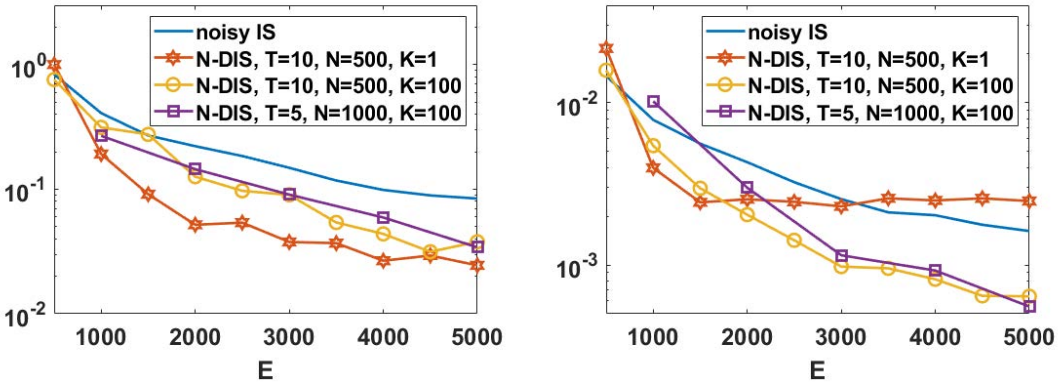
results of the IS schemes on the same noisy target are shown in Figure 6.9.

Improved exploration by surrogates. In this example, the chosen proposal $\varphi(\boldsymbol{\theta}'|\boldsymbol{\theta})$ is not able to explore efficiently the space since the two modes are rather distant. For this reason, the results of PM-MH are much worse than the algorithms that perform several



(a) Estimation of the mean. (b) Estimation of the variance.

Figure 6.6: Relative median squared error in estimation of the mean (left) and variance (right) of the banana pdf with multiplicative exponential noise, by importance sampling schemes.



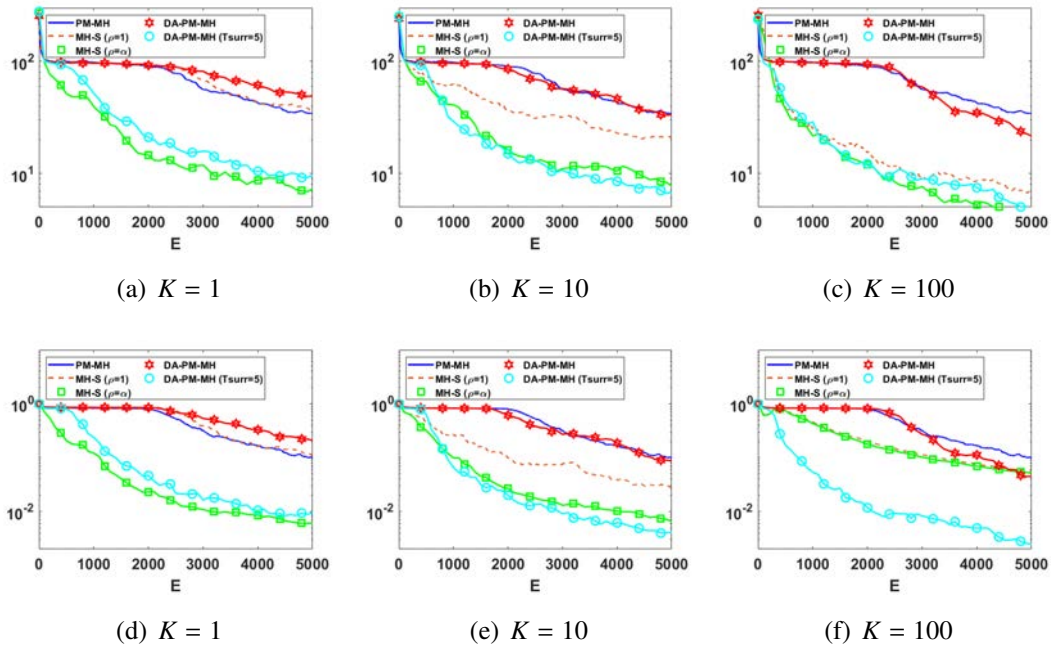
(a) Estimation of the mean. (b) Estimation of the variance.

Figure 6.7: Relative median squared error in estimation of the mean (left) and variance (right) of the banana pdf with rectified additive Gaussian noise, by importance sampling schemes.

steps w.r.t. the surrogate, namely, DA-PM-MH with $T_{\text{surr}} = 5$ and MH-S with $\rho = \alpha$, as can be seen in Figure 6.8. This shows that performing several steps w.r.t. surrogate is beneficial for the exploration and for discovering different modes, specially when the proposal does not propose big jumps. Regarding the results of IS, we see in Figure 6.9 that the use of surrogates improve the performance, but not as much as in the MCMC test, since IS with uniform density already performs very well as compared to PM-MH.

Pathological constructions. In this example, we encounter the negative effect of a bad surrogate construction. In Figures 6.8(a)-(b)-(d)-(e), we see that DA-PM-MH with $T_{\text{surr}} = 1$ performs equal or worse than the baseline technique, i.e., PM-MH. This is probably due to the joint effect of small jumps proposed by $\varphi(\boldsymbol{\theta}|\boldsymbol{\theta}')$ and performing only one step w.r.t. the surrogate, which in turn makes a myopic construction of the surrogate possibly

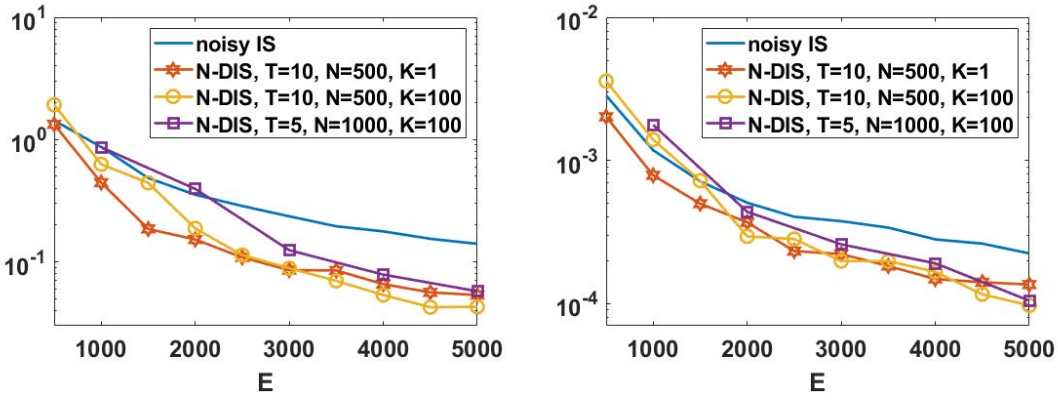
missing one of the modes. This pathological behavior is worst when $K = 1$, but improves as we increase K, matching the performance of PM-MH for $K = 100$.



(a) $K = 1$ (b) $K = 10$ (c) $K = 100$

(d) $K = 1$ (e) $K = 10$ (f) $K = 100$

Figure 6.8: Relative median squared error in estimation of the mean (upper row) and variance (lower row) of the bimodal pdf with multiplicative exponential noise.



(a) Estimation of the mean. (b) Estimation of the variance.

Figure 6.9: Relative median squared error in estimation of the mean (left) and variance (right) of the bimodal pdf with multiplicative exponential noise, by importance sampling schemes.

6.6.3. Double cart pole

We consider a variant of the popular cart-pole system, which is a standard benchmark in RL [31]. In the basic cart-pole environment, the goal is to balance a pole that is hinged on

a cart. The cart is able to move freely along the x-axis. The observations are the position x and velocity $\dot{x}$ of the cart, and the angle α and angular velocity $\dot{\alpha}$ of the pole. The action is continuous and corresponds to the force applied to the cart. The agent receives one point for each iteration that x and α are within some bounds.

We consider here the more challenging variant where another shorter pole is hinged on the cart (see Figure 6.10). Hence, the state vector is $\mathbf{s} = [x, \dot{x}, \alpha_1, \dot{\alpha}_1, \alpha_2, \dot{\alpha}_2]^\top$. The transition p_{env} is deterministic, determined by the evolution of the dynamical system, where each iteration corresponds to 0.02s [65]. We consider the simplest neural network for the policy $a = \pi_{\boldsymbol{\theta}}(\mathbf{s}) = \boldsymbol{\theta}^\top \mathbf{s}$, i.e., a linear policy. Hence, the parameter $\boldsymbol{\theta} \in \mathbb{R}^6$.[12] The return $R(\boldsymbol{\theta}, \boldsymbol{\tau})$ is the number of iterations before any of x, α_1 or α_2 go out of bounds, where $T_{\max} = 1000$. Hence, the maximum return is 1000. Regarding the parameters such as the masses, lengths, friction coefficients, etc., we take the same values as in [31]. At the beginning of each episode, the initial state is obtained by sampling each component uniformly within the following intervals: $x \in [-1.944, 1.944]$, $\dot{x} \in [-1.215, 1.215]$, $\alpha_1 \in [-0.0472, 0.0472]$, $\dot{\alpha}_1 \in [-0.135088, 0.135088]$, $\alpha_2 \in [-0.10472, 0.10472]$ and $\dot{\alpha}_2 \in [-0.135088, 0.135088]$. Note that, in this example, the noisiness comes only from the initial distribution.

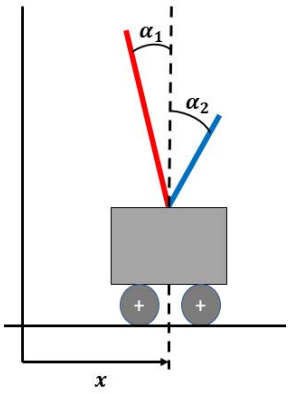


Figure 6.10: Double pole balancing problem.

We consider a realization $\widetilde{m}(\boldsymbol{\theta})$ of $p(\boldsymbol{\theta})$ by simulating one single episode. We first run 10^6 iterations of PM-MH on $\widetilde{m}(\boldsymbol{\theta})$ in order to have a rough estimation (groundtruth) of the marginal histograms w.r.t. we can compare the algorithms. We consider a bounded domain $\boldsymbol{\theta} \in [-60, 60]^6$. We compare two MH-S algorithms and one DA-PM-MH algorithm using again a nearest neighbor surrogate, with $K = 100$. The budget is $E = 10^5$ evaluations. A PM-MH algorithm with the same number of evaluations is also considered. In Figure 6.11, we show the estimated marginal densities. In Table 6.9, we show the MMSE estimations of $\boldsymbol{\theta}$ provided by the different algorithms. We can observe that the compared techniques are able to approximate the groundtruth marginal histograms. However, the DA-PM-MH scheme seems to provide slightly better approximations.

[12] The use of more sophisticated architectures (such as including hidden layers with variable number of hidden units, biases and skip-layers) can produce more effective controllers at the expense of increasing the dimensionality of $\boldsymbol{\theta}$.

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	Exp. return
PM-MH ($T = 10^6$)	-7.1281	-15.0300	5.1756	15.0946	15.4696	4.9734	1000
PM-MH ($T = 10^5$)	-5.6738	-15.7544	3.0080	14.9182	16.3909	6.0570	1000
MH-S ($\rho = 1$)	-6.6351	-10.2346	-1.9859	12.5025	12.8274	6.0455	1000
MH-S ($\rho = \alpha$)	-8.9285	-17.0432	4.0197	13.3249	15.7900	3.9512	1000
DA-PM-MH ($T_{\text{surr}} = 5$)	-5.7748	-17.5469	6.6250	15.9932	17.5892	5.2058	1000

Table 6.9: MMSE estimates for the double cart pole system computed by the different algorithms.

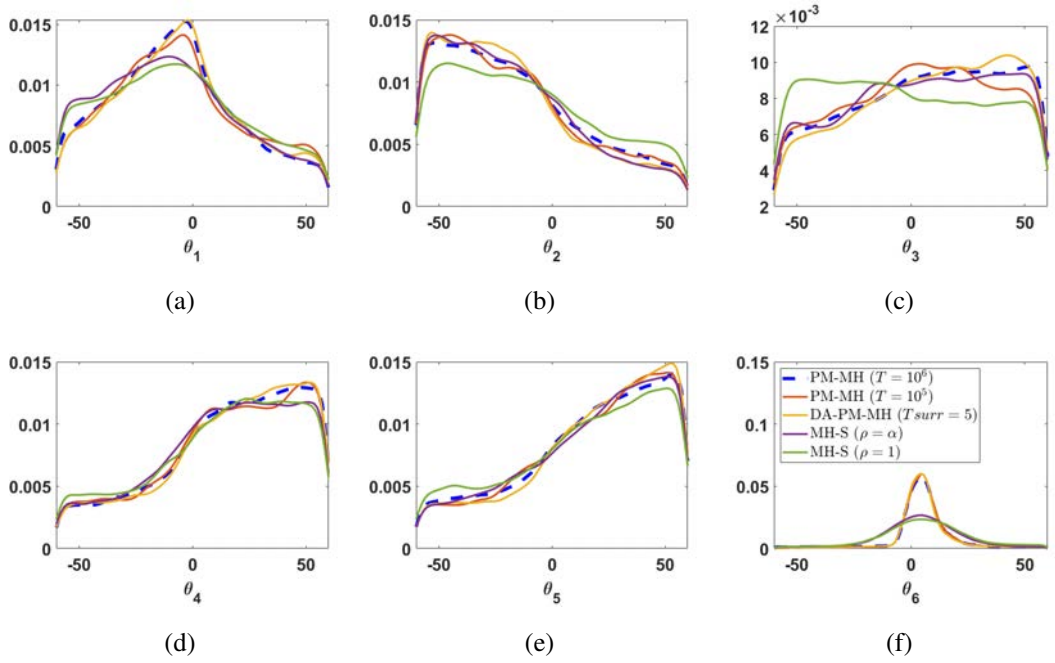


Figure 6.11: Marginal densities of the double cart pole system, obtained by the different algorithms.

6.7. Conclusions

We have provided an overview of Monte Carlo methods which use surrogate models built with regression techniques, for dealing with noisy and costly densities. Indeed, by employing surrogate models, we can avoid the evaluation of expensive true models and perform a smoothing of the noisy realizations. This has important implications for performance in real-world applications.

We have described a general joint framework which encompasses most of the techniques in the literature. We have given a classification of the analyzed techniques in three main families. We have highlighted the connections and differences among the algorithms by means of several explanatory tables and figures. The range of application of the methods have been discussed. Specifically, a detailed description of the likelihood-free approach and the reinforcement learning setting is presented.

Numerical simulations have shown that, generally, the use of surrogates can improve the

performance of the algorithms. Indeed, the surrogate plays the role of an adaptive non-parametric proposal which is adapted using not only the spatial information contained in the samples, but also the noisy evaluations of the target. This increases the efficiency of the corresponding Monte Carlo estimators since it fosters the exploration of the space. On the other hand, pathological constructions of the surrogate, i.e., when the surrogate takes small values in high probability regions of the target pdf, can jeopardize the performance of the algorithms, at least in the first iterations. Furthermore, the correction step in the exact algorithms yields more robust schemes.

6.8. Appendix

6.8.1. Proof for noisy MH algorithm

We provide here a simple proof showing that the invariant density of a MH algorithm using noisy realizations $\widetilde{m}(\boldsymbol{\theta})$ is $m(\boldsymbol{\theta})$ (i.e. a pseudo-marginal MH algorithm). For more details see [5, 6]. Let us consider the acceptance ratio of the noisy MH algorithm

$$r(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_{\text{prop}}) = \frac{\widetilde{m}(\boldsymbol{\theta}_{\text{prop}})\varphi(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}_{\text{prop}})}{\widetilde{m}(\boldsymbol{\theta}_{t-1})\varphi(\boldsymbol{\theta}_{\text{prop}}|\boldsymbol{\theta}_{t-1})}. \tag{6.21}$$

Now, let us rewrite it as

$$r(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_{\text{prop}}) = \frac{\frac{\widetilde{m}(\boldsymbol{\theta}_{\text{prop}})}{m(\boldsymbol{\theta}_{\text{prop}})} m(\boldsymbol{\theta}_{\text{prop}})\varphi(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}_{\text{prop}})}{\frac{\widetilde{m}(\boldsymbol{\theta}_{t-1})}{m(\boldsymbol{\theta}_{t-1})} m(\boldsymbol{\theta}_{t-1})\varphi(\boldsymbol{\theta}_{\text{prop}}|\boldsymbol{\theta}_{t-1})}. \tag{6.22}$$

Define $\lambda = \frac{\widetilde{m}(\boldsymbol{\theta})}{m(\boldsymbol{\theta})}$ as a random variable with pdf given by $g(\lambda|\boldsymbol{\theta})$. Note that $\mathbb{E}[\lambda|\boldsymbol{\theta}] \propto 1$ for any $\boldsymbol{\theta}$. Denoting $\lambda_{\text{prop}} = \frac{\widetilde{m}(\boldsymbol{\theta}_{\text{prop}})}{m(\boldsymbol{\theta}_{\text{prop}})}$ and $\lambda_{t-1} = \frac{\widetilde{m}(\boldsymbol{\theta}_{t-1})}{m(\boldsymbol{\theta}_{t-1})}$, multiplying by $g(\lambda_{\text{prop}}|\boldsymbol{\theta}_{\text{prop}})g(\lambda_{t-1}|\boldsymbol{\theta}_{t-1})$ in both numerator and denominator, and rearranging the terms we see that the acceptance ratio is

$$r(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_{\text{prop}}) = \frac{\lambda_{\text{prop}} m(\boldsymbol{\theta}_{\text{prop}})g(\lambda_{\text{prop}}|\boldsymbol{\theta}_{\text{prop}})\varphi(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}_{\text{prop}})g(\lambda_{t-1}|\boldsymbol{\theta}_{t-1})}{\lambda_{t-1} m(\boldsymbol{\theta}_{t-1})g(\lambda_{t-1}|\boldsymbol{\theta}_{t-1})\varphi(\boldsymbol{\theta}_{\text{prop}}|\boldsymbol{\theta}_{t-1})g(\lambda_{\text{prop}}|\boldsymbol{\theta}_{\text{prop}})}. \tag{6.23}$$

Now, let us define $q_{\text{equiv}}(\boldsymbol{\theta}, \lambda|\boldsymbol{\theta}', \lambda') = g(\lambda|\boldsymbol{\theta})\varphi(\boldsymbol{\theta}|\boldsymbol{\theta}')$ as the equivalent proposal in the joint space $(\boldsymbol{\theta}, \lambda)$. Hence, the ratio is finally expressed as

$$r(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_{\text{prop}}, \lambda_{t-1}, \lambda_{\text{prop}}) = \frac{\lambda_{\text{prop}} m(\boldsymbol{\theta}_{\text{prop}})g(\lambda_{\text{prop}}|\boldsymbol{\theta}_{\text{prop}})q_{\text{equiv}}(\boldsymbol{\theta}_{t-1}, w_{t-1}|\boldsymbol{\theta}_{\text{prop}}, \lambda_{\text{prop}})}{\lambda_{t-1} m(\boldsymbol{\theta}_{t-1})g(\lambda_{t-1}|\boldsymbol{\theta}_{t-1})q_{\text{equiv}}(\boldsymbol{\theta}_{\text{prop}}, w_{\text{prop}}|\boldsymbol{\theta}_{t-1}, \lambda_{t-1})}. \tag{6.24}$$

It can be seen now that the invariant density is proportional to $\lambda \cdot m(\boldsymbol{\theta}) \cdot g(\lambda|\boldsymbol{\theta})$, whose marginal is $\int \lambda\, m(\boldsymbol{\theta})g(\lambda|\boldsymbol{\theta})d\lambda \propto m(\boldsymbol{\theta})$.

6.8.2. Proof for noisy IS

We show that an IS estimator built with noisy realizations $\widetilde{m}(\boldsymbol{\theta})$, converges to expectations w.r.t. $m(\boldsymbol{\theta})$. Let $q(\boldsymbol{\theta})$ denote a proposal pdf, and let

$$\widetilde{Z} = \frac{1}{N}\sum_{i=1}^{N}\frac{\widetilde{m}(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)} = \frac{1}{N}\sum_{i=1}^{N}\widetilde{w}_i, \tag{6.25}$$

be the IS estimator built with noisy realizations, where $\widetilde{w}_i = \frac{\widetilde{m}(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)}$ are the noisy weights, and $\{\boldsymbol{\theta}_i\}_{i=1}^N$ are iid samples from q. The non-noisy IS estimator

$$\widehat{Z} = \frac{1}{N}\sum_{i=1}^{N}\frac{m(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)} = \frac{1}{N}\sum_{i=1}^{N}w_i, \tag{6.26}$$

is an unbiased estimator of $Z = \int m(\boldsymbol{\theta})d\boldsymbol{\theta}$, i.e., $\mathbb{E}[\widehat{Z}] = Z$, converging as $N \to \infty$ at rate $\frac{1}{N}$. We aim to show that $\widetilde{Z}$ is also an unbiased estimator of Z, with greater variance than $\widehat{Z}$, but the same convergence speed, i.e., its variance decreases at $\frac{1}{N}$ rate.

Let $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$ denote the N samples from q. By the law of total expectation, we have that $\mathbb{E}[\widetilde{Z}] = \mathbb{E}\left[\mathbb{E}[\widetilde{Z}|\boldsymbol{\Theta}]\right]$. In the inner expectation, we use the fact the $\widetilde{w}_i$'s are i.i.d., hence

$$\mathbb{E}[\widetilde{Z}|\boldsymbol{\Theta}] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[\widetilde{w}_i|\boldsymbol{\theta}_i] = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{q(\boldsymbol{\theta}_i)}\mathbb{E}[\widetilde{m}(\boldsymbol{\theta}_i)|\boldsymbol{\theta}_i] = \widehat{Z}, \tag{6.27}$$

and

$$\mathbb{E}[\widetilde{Z}] = \mathbb{E}\left[\mathbb{E}[\widetilde{Z}|\boldsymbol{\Theta}]\right] = \mathbb{E}[\widehat{Z}] = Z. \tag{6.28}$$

By the law of total variance, we have that

$$\mathtt{var}[\widetilde{Z}] = \mathbb{E}\left[\mathtt{var}[\widetilde{Z}|\boldsymbol{\Theta}]\right] + \mathtt{var}\left[\mathbb{E}[\widetilde{Z}|\boldsymbol{\Theta}]\right]. \tag{6.29}$$

Using the above result, we have that the second term is

$$\mathtt{var}\left[\mathbb{E}[\widetilde{Z}|\boldsymbol{\Theta}]\right] = \mathtt{var}[\widehat{Z}] = O\left(1/N\right). \tag{6.30}$$

Regarding the first term, we have

$$\begin{aligned}\mathtt{var}[\widetilde{Z}|\boldsymbol{\Theta}] &= \frac{1}{N^2}\sum_{i=1}^{N}\mathtt{var}[\widetilde{w}_i|\boldsymbol{\theta}_i] = \frac{1}{N^2}\sum_{i=1}^{N}\frac{1}{q(\boldsymbol{\theta}_i)^2}\mathtt{var}[\widetilde{m}(\boldsymbol{\theta}_i)|\boldsymbol{\theta}_i] \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\frac{s^2(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)^2}.\end{aligned} \tag{6.31}$$

Assuming that $\frac{s^2(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} < \infty$ for all $\boldsymbol{\theta}$, we have that

$$\mathbb{E}\left[\mathtt{var}[\widetilde{Z}|\boldsymbol{\Theta}]\right] = \frac{1}{N^2}\sum_{i=1}^{N}\mathbb{E}\left[\frac{s^2(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)^2}\right] = \frac{1}{N}\mathbb{E}\left[\frac{s^2(\boldsymbol{\theta})}{q(\boldsymbol{\theta})^2}\right]. \tag{6.32}$$

Hence, we finally have that

$$\begin{aligned}\texttt{var}[\widetilde{Z}] &= \frac{1}{N}\mathbb{E}\left[\frac{s^2(\boldsymbol{\theta})}{q(\boldsymbol{\theta})^2}\right] + \texttt{var}[\widehat{Z}] = O\left(\frac{1}{N}\right) \\ &\geq \texttt{var}[\widehat{Z}]. \end{aligned} \tag{6.33}$$

From this expression, we can deduce that the variance of $\widetilde{Z}$ depends on the mismatch between $q(\boldsymbol{\theta})$ and $s^2(\boldsymbol{\theta})$. Proving that the noisy IS estimator $\widetilde{I} = \frac{1}{N}\sum_{i=1}^{N}\frac{\widetilde{m}(\boldsymbol{\theta})f(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}$ converges to $I = \int f(\boldsymbol{\theta})m(\boldsymbol{\theta})d\boldsymbol{\theta}$ is immediate. Thus, the ratio $\frac{\widetilde{I}}{\widetilde{Z}} = \frac{1}{\sum_{j=1}^{N}\widetilde{w}_j}\sum_{i=1}^{N}\widetilde{w}_i f(\boldsymbol{\theta}_i)$, (i.e. the noisy self-normalized IS estimator) is a consistent estimator of $\frac{\int_{\Theta} f(\boldsymbol{\theta})m(\boldsymbol{\theta})d\boldsymbol{\theta}}{\int_{\Theta} m(\boldsymbol{\theta})d\boldsymbol{\theta}}$.

6.8.3. Analytical expressions of the noise models in illustrative example

Let $\epsilon \sim \mathcal{N}(0, \sigma^2)$. The analytical expressions of $m(\boldsymbol{\theta})$ for the noise models in the illustrative example of Sect. 6.2 are provided here.

Rectified Gaussian. By setting $\widetilde{m}(\boldsymbol{\theta}) = \max(0, p(\boldsymbol{\theta}) + \epsilon)$, then $\widetilde{m}(\boldsymbol{\theta})|\boldsymbol{\theta} \sim \mathcal{N}^{\mathrm{R}}(p(\boldsymbol{\theta}), \sigma^2)$ is a rectified Gaussian random variable, whose mean is

$$m(\boldsymbol{\theta}) = \left[p(\boldsymbol{\theta}) + \sigma\frac{\phi(-p(\boldsymbol{\theta})/\sigma)}{1 - \boldsymbol{\Phi}(-p(\boldsymbol{\theta})/\sigma)}\right]\left[1 - \boldsymbol{\Phi}(-p(\boldsymbol{\theta})/\sigma)\right],$$

where $\phi(\theta)$ and $\boldsymbol{\Phi}(\theta)$ are the pdf and cdf, respectively, of the standard normal distribution.

Folded Gaussian. The random variable $\widetilde{m}(\boldsymbol{\theta}) = |p(\boldsymbol{\theta}) + \epsilon|$ corresponds to a folded Gaussian random variable. We have

$$m(\boldsymbol{\theta}) = \sigma\sqrt{\frac{2}{\pi}}\exp\left(-p^2(\boldsymbol{\theta})/2\sigma^2\right) + p(\boldsymbol{\theta})[1 - 2\boldsymbol{\Phi}(-p(\boldsymbol{\theta})/\sigma)].$$

Bibliography

[1] L. Acerbi. Variational Bayesian Monte Carlo with noisy likelihoods. *arXiv:2006.08655*, 2020.

[2] P. Alquier, N. Friel, R. Everitt, and A. Boland. Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels. *Statistics and Computing*, 26(1-2):29–47, 2016.

[3] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *J. R. Statist. Soc. B*, 72(3):269–342, 2010.

[4] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[5] C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

[6] C. Andrieu and M. Vihola. Convergence properties of pseudo-marginal Markov chain Monte Carlo algorithms. *The Annals of Applied Probability*, 25(2):1030–1077, 2015.

[7] D. V. Arnold. *Noisy optimization with evolution strategies*, volume 8. Springer Science & Business Media, 2012.

[8] M. Balesdent, J. Morio, and J. Marzat. Kriging-based adaptive importance sampling algorithms for rare event estimation. *Structural Safety*, 44:1–10, 2013.

[9] M. Banterle, C. Grazian, A. Lee, and C. P. Robert. Accelerating Metropolis–Hastings algorithms by delayed acceptance. *Foundations of Data Science*, 1(2):103, 2019.

[10] R. Bardenet, A. Doucet, and C. Holmes. On Markov chain Monte Carlo methods for tall data. *The Journal of Machine Learning Research*, 18(1):1515–1557, 2017.

[11] M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

[12] N. Bliznyuk, D. Ruppert, C. Shoemaker, R. Regis, S. Wild, and P. Mugunthan. Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. *Journal of Computational and Graphical Statistics*, 17(2):270–294, 2008.

[13] J. J. Bon, A. Lee, and C. Drovandi. Accelerating sequential Monte Carlo with surrogate likelihoods. *arXiv:2009.03699*, 2020.

[14] F.-X. Briol, C. J. Oates, M. Girolami, M. A. Osborne, and D. Sejdinovic. Probabilistic integration: A role in statistical computation? *Statistical Science*, 34(1):1–22, 2019.

[15] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, and J.-B. Mouret. A survey on policy search algorithms for learning robot controllers in a handful of trials. *IEEE Transactions on Robotics*, 36(2):328–347, 2019.

[16] J. A. Christen and C. Fox. Markov Chain Monte Carlo using an approximation. *Journal of Computational and Graphical statistics*, 14(4):795–810, 2005.

[17] E. Cleary, A. Garbuno-Inigo, S. Lan, T. Schneider, and A. M. Stuart. Calibrate, emulate, sample. *Journal of Computational Physics*, 424:109716, 2021.

[18] P. R. Conrad, Y. M. Marzouk, N. S. Pillai, and A. Smith. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516):1591–1607, 2016.

[19] J.-M. Cornuet, J.-M. Marin, A. Mira, and C. P. Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, 2012.

[20] A. Davis, Y. Marzouk, A. Smith, and N. Pillai. Rate-optimal refinement strategies for local approximation MCMC. *arXiv:2006.00032*, 2020.

[21] M. P. Deisenroth, G. Neumann, and J. Peters. A survey on policy search for robotics. *Foundations and trends in Robotics*, 2(1-2):388–403, 2013.

[22] A. Doucet, M. K Pitt, G. Deligiannidis, and R. Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 102(2):295–313, 2015.

[23] C. C. Drovandi, M. T. Moores, and R. J. Boys. Accelerating pseudo-marginal MCMC using Gaussian processes. *Computational Statistics & Data Analysis*, 118:1–17, 2018.

[24] A. B. Duncan, A. M. Stuart, and M.-T. Wolfram. Ensemble inference methods for models with noisy and expensive likelihoods. *arXiv:2104.03384*, 2021.

[25] R. G. Everitt, A. M. Johansen, E. Rowing, and M. Evdemon-Hogan. Bayesian model comparison with intractable likelihoods. *arXiv*, 1504(06697664):10–1007, 2015.

[26] P. Fearnhead, O. Papaspiliopoulos, G. O. Roberts, and A. Stuart. Random-weight particle filtering of continuous time processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):497–512, 2010.

[27] M. Fielding, D. J. Nott, and S.-Y. Liong. Efficient MCMC schemes for computationally expensive posterior distributions. *Technometrics*, 53(1):16–28, 2011.

[28] V. Gullapalli. *Reinforcement learning and its application to control*. PhD thesis, University of Massachusetts at Amherst, 1992.

[29] M. U. Gutmann and J. Corander. Bayesian optimization for likelihood–free inference of simulator–based statistical models. *Journal of Machine Learning Research*, 2016.

[30] Heikki Haario, Marko Laine, Antonietta Mira, and Eero Saksman. DRAM: efficient adaptive MCMC. *Statistics and computing*, 16(4):339–354, 2006.

[31] V. Heidrich-Meisner and C. Igel. Neuroevolution strategies for episodic reinforcement learning. *Journal of Algorithms*, 64(4):152–168, 2009.

[32] M. Hoffman, A. Doucet, N. De Freitas, and A. Jasra. Trans-dimensional MCMC for Bayesian policy learning. In *NIPS*, volume 20, pages 665–672. Citeseer, 2008.

[33] M. Järvenpää and J. Corander. Approximate Bayesian inference from noisy likelihoods with Gaussian process emulated MCMC. *arXiv:2104.03942*, 2021.

[34] M. Järvenpää, M. U. Gutmann, A. Vehtari, and P. Marttinen. Parallel Gaussian process surrogate Bayesian inference with noisy likelihood evaluations. *Bayesian Analysis*, 16(1):147–178, 2021.

[35] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on evolutionary computation*, 9(3):303–317, 2005.

[36] M. Kanagawa and P. Hennig. Convergence Guarantees for Adaptive Bayesian Quadrature Methods. In *Advances in Neural Information Processing Systems*, pages 6234–6245, 2019.

[37] T. Karvonen, C. J. Oates, and S. Sarkka. A bayes-sard cubature method. In *Advances in Neural Information Processing Systems*, pages 5882–5893, 2018.

[38] Y. M. Ko and E. Byon. Optimal budget allocation for stochastic simulation with importance sampling: exploration vs. replication. *(to appear) IISE Transactions*, pages 1–31, 2021.

[39] J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[40] F. Llorente, L. Martino, D. Delgado-Gómez, and G. Camps-Valls. Deep importance sampling based on regression for model inversion and emulation. *Digital Signal Processing*, 116:103104, 2021.

[41] F. Llorente, L. Martino, V. Elvira, D. Delgado, and J. Lopez-Santiago. Adaptive quadrature schemes for Bayesian inference via active learning. *arXiv:2006.00535*, 2020.

[42] D. Luengo, L. Martino, M. Bugallo, V. Elvira, and S. Särkkä. A survey of Monte Carlo methods for parameter estimation. *EURASIP Journal on Advances in Signal Processing*, 2020:1–62, 2020.

[43] J. M. Marin, P. Pudlo, and M. Sedki. Consistency of the adaptive multiple importance sampling. *arXiv:1211.2548*, 2012.

[44] L. Martino, R. Casarin, F. Leisen, and D. Luengo. Adaptive independent sticky MCMC algorithms. *EURASIP Journal on Advances in Signal Processing*, 2018(1):5, 2018.

[45] L. Martino, V. Elvira, and G. Camps-Valls. The recycling Gibbs sampler for efficient learning. *Digital Signal Processing*, 74:1–13, 2018.

[46] F. J. Medina-Aguayo, A. Lee, and G. O. Roberts. Stability of noisy Metropolis–Hastings. *Statistics and Computing*, 26(6):1187–1211, 2016.

[47] E. Meeds and M. Welling. GPS–ABC: Gaussian process surrogate approximate Bayesian computation. *arXiv:1401.2838*, 2014.

[48] V. Nissen and J. Propach. Optimization with noisy function evaluations. In *International Conference on Parallel Problem Solving from Nature*, pages 159–168. Springer, 1998.

[49] J. Park and M. Haran. Bayesian inference in the presence of intractable normalizing functions. *Journal of the American Statistical Association*, 113(523):1372–1390, 2018.

[50] J. Park and M. Haran. A function emulation approach for doubly intractable distributions. *Journal of Computational and Graphical Statistics*, 29(1):66–77, 2020.

[51] Warren B. Powell. A unified framework for stochastic optimization. *European Journal of Operational Research*, 275(3):795 – 821, 2019.

[52] D. Prangle. Lazy ABC. *Statistics and Computing*, 26(1-2):171–185, 2016.

[53] L. F. Price, C. C. Drovandi, A. Lee, and D. J. Nott. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1):1–11, 2018.

[54] C. E. Rasmussen, J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A.F.M. Smith, and M. West. Gaussian processes to speed up hybrid Monte Carlo for expensive Bayesian integrals. In *Bayesian Statistics 7*, pages 651–659, 2003.

[55] C. P. Robert. Approximate Bayesian computation: A survey on recent results. In *Monte Carlo and Quasi-Monte Carlo Methods*, pages 185–205. Springer, 2016.

[56] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[57] D. B. Rubin. Using the SIR algorithm to simulate posterior distributions. *in Bayesian Statistics 3, ads Bernardo, Degroot, Lindley, and Smith. Oxford University Press, Oxford, 1988.*, 1988.

[58] H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. CRC press, 2005.

[59] C. Sherlock, A. Golightly, and D. A. Henderson. Adaptive, delayed-acceptance MCMC for targets with expensive likelihoods. *Journal of Computational and Graphical Statistics*, 26(2):434–444, 2017.

[60] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[61] D. H. Svendsen, L. Martino, and G. Camps-Valls. Active emulation of computer codes with gaussian processes - application to remote sensing. *Pattern Recognition*, 100:107103, 2020.

[62] V. Tavakol Aghaei, A. Onat, and S. Yıldırım. A Markov chain Monte Carlo algorithm for Bayesian policy search. *Systems Science & Control Engineering*, 6(1):438–455, 2018.

[63] M.-N. Tran, M. Scharth, M. K. Pitt, and R. Kohn. Importance sampling squared for Bayesian inference in latent variable models. *arXiv preprint arXiv:1309.3339*, 2013.

[64] H. Wang and J. Li. Adaptive Gaussian process approximation for Bayesian inference with expensive likelihood functions. *Neural computation*, 30(11):3072–3094, 2018.

[65] A. P. Wieland. Evolving neural network controllers for unstable systems. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume 2, pages 667–673. IEEE, 1991.

[66] R. Wilkinson. Accelerating ABC methods using Gaussian processes. In *Artificial Intelligence and Statistics*, pages 1015–1023. PMLR, 2014.

[67] H. Ying, K. Mao, and K. Mosegaard. Moving Target Monte Carlo. *arXiv:2003.04873*, 2020.

[68] J. Zhang and A. A. Taflanidis. Accelerating MCMC via kriging-based adaptive independent proposals and delayed rejection. *Computer Methods in Applied Mechanics and Engineering*, 355:1124–1147, 2019.

7. OPTIMALITY IN NOISY IMPORTANCE SAMPLING

In *Signal Processing*, Volume 194, 108455 (2022)

Fernando Llorente*, Luca Martino⋆, Jesse Read†, David Delgado–Gómez*

* Universidad Carlos III de Madrid, Leganés, Spain.

⋆ Universidad Rey Juan Carlos, Fuenlabrada, Spain.

† École Polytechnique, Palaiseau, France.

Abstract

Many applications in signal processing and machine learning require the study of probability density functions (pdfs) that can only be accessed through noisy evaluations. In this work, we analyze the noisy importance sampling (IS), i.e., IS working with noisy evaluations of the target density. We present the general framework and derive optimal proposal densities for noisy IS estimators. The optimal proposals incorporate the information of the variance of the noisy realizations, proposing points in regions where the noise power is higher. We also compare the use of the optimal proposals with previous optimality approaches considered in a noisy IS framework.

Keywords: Bayesian Inference; Noisy Monte Carlo; Pseudo-marginal Metropolis-Hastings; Noisy IS.

7.1. Introduction

A wide range of modern applications, especially in Bayesian inference framework [22], require the study of probability density functions (pdfs) which can be evaluated stochastically, i.e., only noisy evaluations can be obtained [16, 30, 1, 21]. For instance, this is the case of the pseudo-marginal approaches and doubly intractable posteriors [4, 24], approximate Bayesian computation (ABC) and likelihood-free schemes [25, 17], where the target density cannot be computed in closed-form.
The noisy scenario also appears naturally when mini-batches of data are employed instead of considering the complete likelihood of huge amounts of data [6, 26]. More recently, the analysis of noisy functions of densities is required in reinforcement learning (RL), especially in direct policy search which is an important branch of RL, with applications in robotics [11, 8]. The topic of inference in noisy settings (or where a function is known with a certain degree of uncertainty) is also of interest in the inverse problem literature, such as in the calibration of expensive computer codes [14, 7]. This is also the case when the construction of an *emulator* is considered, as a surrogate model [1, 29, 20].

In this work, we study the importance sampling (IS) scheme under noisy evaluations of the target pdf. The noisy IS scenario has been already analyzed in the literature [16,

30, 15]. In the context of optimization, some theoretical results can be found [2]. In the sequential framework, IS schemes with random weights can be found and have been studied in different works [16, 10, 9, 23]. We provide the optimal proposal densities for different noisy scenarios, including also the case of integrals involving vector-valued functions. Moreover, we discuss the convergence and variance of the estimators in a general setting. We consider a different approach with respect to other studies in the literature [30, 12]. In those works, the authors analyzed the trade-off between decreasing the noise power (by increasing the number of auxiliary samples) and increasing the total number of samples in the IS estimators. Here, this information is encompassed within the optimal proposal density, which plays a similar role to an acquisition function in active learning [20, 29]. This is information is relevant, especially if the noisy evaluations are also costly to obtain.

7.2. Background

7.2.1. Bayesian inference

In many applications, we aim at inferring a variable of interest given a set of observations or measurements. Let us denote the variable of interest by $\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^{d_x}$, and let $\mathbf{y} \in \mathbb{R}^{d_y}$ be the observed data. The posterior pdf is then

$$\bar{p}(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{Z(\mathbf{y})}, \tag{7.1}$$

where $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf, and $Z(\mathbf{y})$ is the model evidence (a.k.a. marginal likelihood) which is a useful quantity in model selection problems [19]. For simplicity, in the following, we skip the dependence on $\mathbf{y}$ in $\bar{p}(\mathbf{x}) = \bar{p}(\mathbf{x}|\mathbf{y})$ and $Z = Z(\mathbf{y})$. Generally, Z is unknown, so we are able to evaluate the unnormalized target function, i.e., the numerator on the right hand side of Eq. (7.1),

$$p(\mathbf{x}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}). \tag{7.2}$$

The analytical study of the posterior density $\bar{p}(\mathbf{x}) \propto p(\mathbf{x})$ is unfeasible, so that numerical approximations are required [27, 22].

7.2.2. Noisy framework

Generally, we desire to approximate the unnormalized density $p(\mathbf{x})$, $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$, and the corresponding normalizing constant Z, using Monte Carlo methods. The unnormalized density $p(\mathbf{x})$ can represent a posterior density in a Bayesian inference problem, as described above. We assume that, for any $\mathbf{x}$, we cannot evaluate $p(\mathbf{x})$ exactly, but we only have access to a related noisy realization. Moreover, in many applications, obtaining such a noisy realization may be expensive. Hence, analyzing in which $\mathbf{x}$ we require a noisy

realization of $p(\mathbf{x})$ is an important problem, which is related to the concept of *optimality* that we consider below.
In the following, we introduce a concise mathematical formalization of the noisy scenario. This simple framework contains real application scenarios, such as latent variable models [30] (see example 4 in Sect. 7.4.1), likelihood-free inference setting [25], doubly intractable posteriors [24], mini batch-based inference [6]. More specifically, we assume to have access to a noisy realization related to $p(\mathbf{x})$, i.e.,

$$\widetilde{m}(\mathbf{x}) = H(p(\mathbf{x}), \epsilon), \tag{7.3}$$

where H is a non-linear transformation involving $p(\mathbf{x})$ and ϵ, that is some noise perturbation. Thus, *for a fixed value* $\boldsymbol{x}$, $\widetilde{m}(\mathbf{x})$ is a random variable with

$$m(\mathbf{x}) = \mathbb{E}[\widetilde{m}(\mathbf{x})], \quad s(\mathbf{x})^2 = \mathrm{Var}[\widetilde{m}(\mathbf{x})], \tag{7.4}$$

for some *mean function*, $m(\mathbf{x})$, and *variance function*, $s(\mathbf{x})^2$. The assumption that $\widetilde{m}(\mathbf{x})$ must be strictly positive is important in practice [16, 12].

Noise power. In some applications, it is also possible to control the noise power $s(\mathbf{x})^2$, for instance by adding/removing data to the mini-batches (e.g., in the context of Big Data) [6], increasing the number of auxiliary samples in latent variables models [4], or interacting with an environment over longer/shorter periods of time (e.g., in reinforcement learning) [11].

Unbiased scenario and related cases. The scenario where $m(\mathbf{x}) = p(\mathbf{x})$ appears naturally in some applications (such as in the estimation of latent variable and stochastic volatility models in statistics [30, 3]; or in the context of optimal filtering of partially observed stochastic processes [15]), or it is often assumed as a pre-established condition by the authors [30, 16]. In some other scenarios, the noisy realizations are known to be unbiased estimates of some transformation of $p(\mathbf{x})$, e.g., of $\log p(\mathbf{x})$ [18, 13]. This situation can be encompassed by the following special case. If we consider an additive perturbation,

$$\widetilde{m}(\mathbf{x}) = G\left(p(\mathbf{x})\right) + \epsilon, \qquad \text{with } \mathbb{E}[\epsilon] = 0, \tag{7.5}$$

we have $m(\mathbf{x}) = \mathbb{E}[\widetilde{m}(\mathbf{x})] = G\left(p(\mathbf{x})\right)$, where $G(\cdot) : \mathbb{R} \to \mathbb{R}$. If G is known and invertible, we have $p(\mathbf{x}) = G^{-1}\left(m(\mathbf{x})\right)$.
Generally, we can state that $m(\mathbf{x})$ always contains statistical information related to $p(\mathbf{x})$. The subsequent use of $m(\mathbf{x})$ depends on the specific application. Thus, we study the mean function $m(\mathbf{x})$. Hence, our goal is to approximate efficiently integrals involving $m(\mathbf{x})$, i.e.,

$$\mathbf{I} = \frac{1}{\bar{Z}} \int_{\mathcal{X}} \mathbf{f}(\mathbf{x}) m(\mathbf{x}) d\mathbf{x}, \quad \bar{Z} = \int_{\mathcal{X}} m(\mathbf{x}) d\mathbf{x}, \tag{7.6}$$

where $\mathbf{f}(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^{d_f}$ and $\mathbf{I} = [I_1, \ldots, I_{d_f}]^\top \in \mathbb{R}^{d_f}$ denotes the vector of integrals of interest. Note that, in the unbiased case $m(\mathbf{x}) = p(\mathbf{x})$, we have $\bar{Z} = Z$. An integral involving $m(\mathbf{x})$ can be approximated employing a cloud of random samples using the noisy realizations $\widetilde{m}(\mathbf{x})$ via Monte Carlo methods.

7.3. Noisy Importance Sampling

In a non-noisy IS scheme, a set of samples is drawn from a proposal density $q(\mathbf{x})$. Then each sample is weighted according to the ratio $\frac{p(\mathbf{x})}{q(\mathbf{x})}$. A noisy version of importance sampling can be obtained when we substitute the evaluations of $p(\mathbf{x})$ with noisy realizations of $\widetilde{m}(\mathbf{x})$. See Table 7.1 and note that the importance weights w_n in Eq. (7.9) are computed using the noisy realizations. Below, we show that

$$\widehat{Z} = \frac{1}{N}\sum_{n=1}^{N} w_n, \tag{7.7}$$

is an unbiased estimator of $\bar{Z}$, and

$$\widehat{\mathbf{I}}_{\text{std}} = \frac{1}{N\bar{Z}}\sum_{n=1}^{N} w_n \mathbf{f}(\mathbf{x}_n), \qquad \widehat{\mathbf{I}}_{\text{self}} = \frac{1}{N\widehat{Z}}\sum_{n=1}^{N} w_n \mathbf{f}(\mathbf{x}_n), \tag{7.8}$$

are consistent estimators of $\mathbf{I}$. The estimator $\widehat{\mathbf{I}}_{\text{std}}$ requires the knowledge of $\bar{Z}$, that is not needed in the so-called self-normalized estimator, $\widehat{\mathbf{I}}_{\text{self}}$.

Table 7.1: Noisy importance sampling algorithm

1. **Inputs:** Proposal distribution $q(\mathbf{x})$.
2. For $n = 1, \ldots, N$:
   (a) Sample $\mathbf{x}_n \sim q(\mathbf{x})$ and obtain one realization $\widetilde{m}(\mathbf{x}_n)$.
   (b) Compute
   $$w_n = \frac{\widetilde{m}(\mathbf{x}_n)}{q(\mathbf{x}_n)} \tag{7.9}$$
4 **Outputs:** the weighted samples $\{\mathbf{x}_n, w_n\}_{n=1}^{N}$.

Theorem 3. *The estimators above constructed from the output of noisy IS converge to expectations under $m(\boldsymbol{x})$. More specifically, we have $\widehat{Z}$ and $\widehat{\mathbf{I}}_{std}$ are unbiased estimators of $\bar{Z}$ and $\mathbf{I}$ respectively, and $\widehat{\mathbf{I}}_{self}$ is a consistent estimator of $\mathbf{I}$. Moreover, these estimators have higher variance than their non-noisy counterparts.*

Proof. Here, we provide a simple proof of convergence by applying iterated conditional expectations. Equivalently, the correctness of the approach can be proved by using an extended space view (see, e.g., [15, 30]).
Let $\mathbf{x}_{1:N} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ denote the N samples from q. By the law of total expectation, we have that $\mathbb{E}[\widehat{Z}] = \mathbb{E}\left[\mathbb{E}[\widehat{Z}|\mathbf{x}_{1:N}]\right]$. In the inner expectation, we use the fact the w_i's are i.i.d., hence

$$\mathbb{E}[\widehat{Z}|\mathbf{x}_{1:N}] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[w_i|\mathbf{x}_i] = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{q(\mathbf{x}_i)}\mathbb{E}[\widetilde{m}(\mathbf{x}_i)|\mathbf{x}_i] = \frac{1}{N}\sum_{i=1}^{N}\frac{m(\mathbf{x}_i)}{q(\mathbf{x}_i)} = \widetilde{Z},$$

where $\widetilde{Z}$ is the non-noisy IS estimator of $\bar{Z}$, which is also unbiased, i.e.,

$$\mathbb{E}[\widehat{Z}] = \mathbb{E}\left[\mathbb{E}[\widehat{Z}|\mathbf{x}_{1:N}]\right] = \mathbb{E}[\widetilde{Z}] = \bar{Z}.$$

Therefore, $\widehat{Z}$ is an unbiased estimator of $\bar{Z} = \int_{\mathcal{X}} m(\mathbf{x})d\mathbf{x}$, i.e., $\mathbb{E}[\widehat{Z}] = \bar{Z}$. Moreover, we show below that $\text{Var}[\widehat{Z}]$ decreases to zero as $N \to \infty$. Hence, $\widehat{Z}$ is a consistent estimator of $\bar{Z}$. Now, with the same arguments, we can prove that the estimator $\widehat{\mathbf{E}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\widetilde{m}(\mathbf{x})\mathbf{f}(\mathbf{x})}{q(\mathbf{x})}$ is also unbiased and converges to $\mathbf{E} = \int_{\mathcal{X}} \mathbf{f}(\mathbf{x})m(\mathbf{x})d\mathbf{x}$. Thus, both the estimator $\widehat{\mathbf{I}}_{\text{std}}$, and the ratio

$$\widehat{\mathbf{I}} = \frac{1}{\widehat{Z}}\widehat{\mathbf{E}} = \frac{1}{\sum_{j=1}^{N} w_j}\sum_{i=1}^{N} w_i \mathbf{f}(\mathbf{x}_i),$$

which is the noisy self-normalized IS estimator $\widehat{\mathbf{I}}_{\text{self}}$ in Eq. (7.8), are consistent estimators of

$$\mathbf{I} = \frac{\int_{\mathcal{X}} \mathbf{f}(\mathbf{x})m(\mathbf{x})d\mathbf{x}}{\int_{\mathcal{X}} m(\mathbf{x})d\mathbf{x}} = \frac{1}{\bar{Z}}\int_{\mathcal{X}} \mathbf{f}(\mathbf{x})m(\mathbf{x})d\mathbf{x},$$

given in Eq. (7.6).

□

Variance of $\widehat{Z}$. By the law of total variance, we have that

$$\text{Var}[\widehat{Z}] = \mathbb{E}\left[\text{Var}[\widehat{Z}|\mathbf{x}_{1:N}]\right] + \text{Var}\left[\mathbb{E}[\widehat{Z}|\mathbf{x}_{1:N}]\right].$$

In a non-noisy scenario, i.e., in a non-noisy IS setting, the first term is null. Using the fact that $\widehat{Z}$ is unbiased, we have that the second term is

$$\text{Var}\left[\mathbb{E}[\widehat{Z}|\mathbf{x}_{1:N}]\right] = \text{Var}[\widetilde{Z}] = O(1/N).$$

Regarding the first term, we have

$$\text{Var}[\widehat{Z}|\mathbf{x}_{1:N}] = \frac{1}{N^2}\sum_{i=1}^{N} \text{Var}[w_i|\mathbf{x}_i] = \frac{1}{N^2}\sum_{i=1}^{N} \frac{1}{q(\mathbf{x}_i)^2}\text{Var}[\widetilde{m}(\mathbf{x}_i)|\mathbf{x}_i] = \frac{1}{N^2}\sum_{i=1}^{N} \frac{s(\mathbf{x}_i)^2}{q(\mathbf{x}_i)^2}.$$

Assuming that $\frac{s(\mathbf{x})^2}{q(\mathbf{x})^2} < \infty$ for all $\mathbf{x}$, we have that

$$\mathbb{E}\left[\text{Var}[\widehat{Z}|\mathbf{x}_{1:N}]\right] = \frac{1}{N^2}\sum_{i=1}^{N} \mathbb{E}\left[\frac{s(\mathbf{x}_i)^2}{q(\mathbf{x}_i)^2}\right] = \frac{1}{N}\mathbb{E}\left[\frac{s(\mathbf{x})^2}{q(\mathbf{x})^2}\right], \quad \text{where} \quad \mathbf{x} \sim q(\mathbf{x}).$$

Hence, we finally have that

$$\text{Var}[\widehat{Z}] = \frac{1}{N}\mathbb{E}\left[\frac{s(\mathbf{x})^2}{q(\mathbf{x})^2}\right] + \text{Var}[\widetilde{Z}] \geq \text{Var}[\widetilde{Z}]. \tag{7.10}$$

Therefore, $\widehat{Z}$ has a greater variance than $\widetilde{Z}$, but the same convergence speed, i.e., its variance decreases at $\frac{1}{N}$ rate. Proving that $\widehat{\mathbf{E}}$ has greater variance than its non-noisy version is straightforward.

7.4. Optimal Proposal Density in Noisy IS

In this section, we derive the optimal proposals for the noisy IS estimators $\widehat{Z}$, $\widehat{\mathbf{I}}_{\text{std}}$ and $\widehat{\mathbf{I}}_{\text{self}}$.

7.4.1. Optimal proposal for $\widehat{Z}$

We can rewrite the variance of $\widehat{Z}$ in Eq. (7.10) as

$$\text{Var}[\widehat{Z}] = \frac{1}{N}\mathbb{E}\left[\frac{m(\mathbf{x})^2 + s(\mathbf{x})^2}{q(\mathbf{x})^2}\right] - \frac{1}{N}\bar{Z}^2.$$

By Jensen's inequality, the first term is bounded below by

$$\mathbb{E}\left[\frac{m(\mathbf{x})^2 + s(\mathbf{x})^2}{q(\mathbf{x})^2}\right] \geq \left(\mathbb{E}\left[\frac{\sqrt{m(\mathbf{x})^2 + s(\mathbf{x})^2}}{q(\mathbf{x})}\right]\right)^2.$$

The minimum variance $\text{V}_{\text{min}} = \min_q \text{Var}[\widehat{Z}]$ is thus attained at

$$q_{\text{opt}}(\mathbf{x}) \propto \sqrt{m(\mathbf{x})^2 + s(\mathbf{x})^2}, \tag{7.11}$$

Note that, for finite N, V_{min} is always greater than 0, specifically,

$$\text{V}_{\text{min}} = \frac{1}{N}\left[\int_{\mathcal{X}} \sqrt{m(\mathbf{x})^2 + s(\mathbf{x})^2} d\mathbf{x}\right]^2 - \frac{1}{N}\bar{Z}^2. \tag{7.12}$$

Hence, differently from the non-noisy setting, in noisy IS the optimal proposal does not provide an estimator with null variance. If $s(\mathbf{x}) = 0$ for all $\mathbf{x}$, then we come back to the non-noisy scenario and $\text{V}_{\text{min}} = \frac{1}{N}\left[\int_{\mathcal{X}} m(\mathbf{x}) d\mathbf{x}\right]^2 - \frac{1}{N}\bar{Z}^2 = 0$. Note that the variance of using $q(\mathbf{x}) = \frac{1}{\bar{Z}} m(\mathbf{x})$ is

$$\text{V}_{\text{sub-opt}} = \frac{\bar{Z}}{N}\int_{\mathcal{X}} \frac{m(\mathbf{x})^2 + s(\mathbf{x})^2}{m(\mathbf{x})} d\mathbf{x} - \frac{1}{N}\bar{Z}^2 = \frac{\bar{Z}}{N}\int_{\mathcal{X}} \frac{s(\mathbf{x})^2}{m(\mathbf{x})} d\mathbf{x}. \tag{7.13}$$

In the following, we show several examples of noise models and their corresponding optimal proposal densities.

Example 1. Let us consider a Bernoulli-type noise where $\widetilde{m}(\mathbf{x}) = p_{\max}\epsilon$, where $\epsilon \sim \text{Bernoulli}\left(\frac{p(\mathbf{x})}{p_{\max}}\right)$, and $p_{\max} = \max p(\mathbf{x})$. Then, we have

$$m(\mathbf{x}) = p(\mathbf{x}), \qquad s(\mathbf{x})^2 = p(\mathbf{x})[p_{\max} - p(\mathbf{x})].$$

Replacing in Eq. (7.11), the optimal proposal density in this case is

$$q_{\text{opt}}(\mathbf{x}) \propto p(\mathbf{x})\sqrt{1 + [p_{\max} - p(\mathbf{x})]^2}. \tag{7.14}$$

Example 2. Let us consider $\widetilde{m}(\mathbf{x}) = |p(\mathbf{x}) + \epsilon|$, with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. In this scenario, the random variable $\widetilde{m}(\mathbf{x})$ corresponds to a folded Gaussian random variable. We have

$$m(\mathbf{x}) = \sigma\sqrt{\frac{2}{\pi}}\exp\left(-p^2(\mathbf{x})/2\sigma^2\right) + p(\mathbf{x})[1 - 2\mathbf{\Phi}(-p(\mathbf{x})/\sigma)],$$
$$s(\mathbf{x})^2 = p(\mathbf{x})^2 + \sigma^2 - m(\mathbf{x})^2,$$

where $\mathbf{\Phi}(\mathbf{x})$ is the cumulative function of the standard Gaussian distribution. Then,

$$q_{\text{opt}}(\mathbf{x}) \propto \sqrt{p(\mathbf{x})^2 + \sigma^2}. \tag{7.15}$$

Example 3. Let us consider a multiplicative noise $\widetilde{m}(\mathbf{x}) = e^{\epsilon} p(\mathbf{x})$ with $\mathbb{E}[\epsilon] = 0$, hence

$$m(\mathbf{x}) = p(\mathbf{x})\mathbb{E}[e^{\epsilon}] \propto p(\mathbf{x}), \qquad s(\mathbf{x})^2 = p(\mathbf{x})^2 \text{Var}[e^{\epsilon}].$$

If we denote $A = \mathbb{E}[e^{\epsilon}]$ and $\sigma^2 = \text{Var}[e^{\epsilon}]$, then $m(\mathbf{x}) = Ap(\mathbf{x})$ and $s(\mathbf{x})^2 = \sigma^2 p^2(\mathbf{x})$. In this case, the optimal proposal coincides with the optimal one in the non-noisy setting, since

$$q_{\text{opt}}(\mathbf{x}) \propto \sqrt{A^2 p^2(\mathbf{x}) + \sigma^2 p^2(\mathbf{x})} = p(\mathbf{x}) \sqrt{A^2 + \sigma^2} \propto p(\mathbf{x}). \tag{7.16}$$

Example 4. In latent variable models, the noisy realization corresponds to the product of d_y independent IS estimators, each built from R auxiliary samples. With d_y large enough, the distribution of this realization is approximately lognormal, i.e.,

$$\widetilde{m}(\mathbf{x}) \sim \log \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x})),$$

where $\mu(\mathbf{x}) = \log p(\mathbf{x}) - \frac{\gamma^2(\mathbf{x})}{2R}$ and $\sigma^2(\mathbf{x}) = \frac{\gamma^2(\mathbf{x})}{R}$, for some function $\gamma^2(\mathbf{x})$ [30, 12]. Equivalently, they write $\widetilde{m}(\mathbf{x}) = p(\mathbf{x})e^{\epsilon}$, where $\epsilon \sim \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$. Hence,

$$m(\mathbf{x}) = p(\mathbf{x}), \qquad s(\mathbf{x})^2 = (e^{\gamma^2(\mathbf{x})/R} - 1)p(\mathbf{x})^2,$$

and the optimal proposal is

$$q_{\text{opt}}(\mathbf{x}) \propto p(\mathbf{x}) e^{\frac{\gamma^2(\mathbf{x})}{2R}}. \tag{7.17}$$

This example is related with the cases studied in [30, 12].

7.4.2. Optimal proposal for $\widehat{\mathbf{I}}_{std}$

We have already seen that the optimal proposal that minimizes the variance of $\widehat{Z}$ is $q_{\text{opt}}(\mathbf{x}) \propto \sqrt{m(\mathbf{x})^2 + s(\mathbf{x})^2}$. Let us consider now the estimator $\widehat{\mathbf{I}}_{\text{std}}$. Note that this estimator assumes we can evaluate $\bar{Z} = \int_{\mathcal{X}} m(\mathbf{x})d\mathbf{x}$. Since we are considering a vector-valued function, the estimator has d_f components $\widehat{\mathbf{I}}_{\text{std}} = [\widehat{I}_{\text{std},1} \dots \widehat{I}_{\text{std},d_f}]^\top$, and $\text{Var}[\widehat{\mathbf{I}}_{\text{std}}]$ corresponds to a $d_f \times d_f$ covariance matrix. We aim to find the proposal that minimizes the sum of diagonal variances. From the results of the previous section, it is straightforward to show that the variance of the p-th component is

$$\begin{aligned}\text{Var}[\widehat{I}_{\text{std},p}] &= \text{Var}[\widetilde{I}_{\text{std},p}] + \frac{1}{N\bar{Z}^2}\mathbb{E}\left[\frac{f_p(\mathbf{x})^2 s(\mathbf{x})^2}{q(\mathbf{x})^2}\right] \\ &= \frac{1}{N\bar{Z}^2}\mathbb{E}\left[\frac{f_p(\mathbf{x})^2(m(\mathbf{x})^2 + s(\mathbf{x})^2)}{q(\mathbf{x})^2}\right] - \frac{1}{N\bar{Z}^2} I_p^2,\end{aligned}$$

where $f_p(\mathbf{x})$ and I_p are respectively the p-th components of $\mathbf{f}(\mathbf{x})$ and $\mathbf{I}$, and $\widetilde{I}_{\text{std},p}$ denotes the non-noisy estimator (i.e. using $m(\mathbf{x})$ instead of $\widetilde{m}(\mathbf{x})$). Thus,

$$\sum_{p=1}^{d_f} \text{Var}[\widehat{I}_{\text{std},p}] = \frac{1}{N\bar{Z}^2}\mathbb{E}\left[\frac{\sum_{p=1}^{d_f} f_p(\mathbf{x})^2(m(\mathbf{x})^2 + s(\mathbf{x})^2)}{q(\mathbf{x})^2}\right] - \frac{1}{N\bar{Z}^2}\sum_{p=1}^{d_f} I_p^2.$$

By Jensen's inequality, we have

$$\mathbb{E}\left[\frac{\sum_{p=1}^{d_f} f_p(\mathbf{x})^2(m(\mathbf{x})^2+s(\mathbf{x})^2)}{q(\mathbf{x})^2}\right] \geq \left(\mathbb{E}\left[\frac{\sqrt{m(\mathbf{x})^2+s(\mathbf{x})^2}\,\|\mathbf{f}(\mathbf{x})\|_2}{q(\mathbf{x})}\right]\right)^2,$$

where $\|\mathbf{f}(\mathbf{x})\|_2$ denotes the euclidean norm. The equality holds if and only if $\frac{\sqrt{m(\mathbf{x})^2+s(\mathbf{x})^2}\|\mathbf{f}(\mathbf{x})\|_2}{q(\mathbf{x})}$ is constant. Hence, the optimal proposal is

$$q_{\text{opt}}(\mathbf{x}) \propto \|\mathbf{f}(\mathbf{x})\|_2 \sqrt{m(\mathbf{x})^2+s(\mathbf{x})^2}. \tag{7.18}$$

7.4.3. Optimal proposal for $\widehat{\mathbf{I}}_{self}$

Let us consider the case of the self-normalized estimator $\widehat{\mathbf{I}}_{\text{self}}$. Recall that $\widehat{\mathbf{I}}_{\text{self}} = \frac{\widehat{\mathbf{E}}}{\widehat{Z}}$, where $\widehat{\mathbf{E}}$ denotes the noisy estimator of $\mathbf{E} = \int_{\mathcal{X}} \mathbf{f}(\mathbf{x})m(\mathbf{x})d\mathbf{x}$, so that we are considering ratios of estimators. Again, we aim to find the proposal that minimizes the variance of the vector-valued estimator $\widehat{\mathbf{I}}_{\text{self}}$. When N is large enough, the variance of p-th ratio is approximated as [28],

$$\text{Var}[\widehat{I}_{\text{self},p}] = \text{Var}\left[\frac{\widehat{E}_p}{\widehat{Z}}\right] \approx \frac{1}{\bar{Z}^2}\text{Var}[\widehat{E}_p] - 2\frac{E_p}{Z}\text{Cov}[\widehat{E}_p,\widehat{Z}] + \frac{E_p^2}{\bar{Z}^4}\text{Var}[\widehat{Z}],$$

where E_p is the p-th component of $\mathbf{E}$, and

$$\begin{aligned}
\text{Var}[\widehat{E}_p] &= \frac{1}{N}\mathbb{E}\left[\frac{f_p(\mathbf{x})^2(m(\mathbf{x})^2+s(\mathbf{x})^2)}{q(\mathbf{x})^2}\right] - \frac{1}{N}E_p^2,\\
\text{Var}[\widehat{Z}] &= \frac{1}{N}\mathbb{E}\left[\frac{m(\mathbf{x})^2+s(\mathbf{x})^2}{q(\mathbf{x})^2}\right] - \frac{1}{N}\bar{Z}^2,\\
\text{Cov}[\widehat{E}_p,\widehat{Z}] &= \frac{1}{N}\mathbb{E}\left[\frac{f_p(\mathbf{x})(m(\mathbf{x})^2+s(\mathbf{x})^2)}{q(\mathbf{x})^2}\right] - \frac{1}{N}E_p\bar{Z}.
\end{aligned}$$

The first two results have been already obtained in the previous sections. The third result is given in Appendix 7.7.1. The sum of the variances is thus

$$\sum_{p=1}^{d_f}\text{Var}[\widehat{I}_{\text{self},p}] \approx \frac{1}{N\bar{Z}^2}\mathbb{E}\left[\frac{(m(\mathbf{x})^2+s(\mathbf{x})^2)\sum_{p=1}^{d_f}(f_p(\mathbf{x})-I_p)^2}{q(\mathbf{x})^2}\right].$$

By Jensen's inequality, we can derive that the optimal proposal is

$$q_{\text{opt}}(\mathbf{x}) \propto \|\mathbf{f}(\mathbf{x})-\mathbf{I}\|_2 \sqrt{m(\mathbf{x})^2+s(\mathbf{x})^2}. \tag{7.19}$$

Relationship with active learning. The optimal density $q_{opt}(\mathbf{x})$ can be interpreted as an *acquisition density*, suggesting the regions of the space which require more number of acquisitions of the realizations $\widetilde{m}(\mathbf{x})$. Namely, $q_{opt}(\mathbf{x})$ plays a role similar to an acquisition function in active learning. This is information is relevant, especially if the noisy evaluations are also costly to obtain.

7.4.4. Connection with other types of optimality

Here, we discuss another approach for optimality in noisy IS and connect it with our work. Other related works, in Monte Carlo and noisy optimization literature, focus on the trade-off between accuracy/noisiness and computational cost [30, 12, 5]. In those settings, it is assumed that one can control the variance $s(\mathbf{x})^2$ of the noisy realizations $\widetilde{m}(\mathbf{x})$. Clearly, taking samples with higher accuracy, i.e. small variance $s(\mathbf{x})^2$, is beneficial since it decrease the magnitude of the terms $\mathbb{E}\left[\frac{s(\mathbf{x})^2}{m(\mathbf{x})^2}\right]$ and $\mathbb{E}\left[\frac{f_p(\mathbf{x})^2 s(\mathbf{x})^2}{m(\mathbf{x})^2}\right]$, which are responsible for the efficiency loss in the estimators, due to the presence of noise. However, taking accurate estimates implies increased computational cost, hence one must reduce the number of samples N, which affect the overall Monte Carlo variance. This trade-off have been investigated in both MCMC and IS frameworks [30, 12].

Let R denote the number of auxiliary samples employed to reduce the variance of the noisy realizations. Namely, greater R implies greater accuracy but also greater cost. Moreover, this number could depend on $\mathbf{x}$, i.e., $R(\mathbf{x}) : \mathcal{X} \to \mathbb{N}^+\backslash\{0\}$. Then, the goal is to obtain the optimal function $R(\mathbf{x})$ by balancing the decrease in variance with the extra computational cost (see, e.g., Sections 3.3, 3.4 and 5 of [30]). Namely, in this different approach, they try to reduce $s(\mathbf{x})^2$ at certain $\mathbf{x}$ increasing the value of $R(\mathbf{x})$, instead of using an optimal proposal pdf for the noisy scenario. On the contrary, in this work we have considered the use of optimal proposal pdfs and that $s(\mathbf{x})^2$ is not tuned by the user, which means that R is arbitrary and set constant for all $\mathbf{x}$.

7.5. Numerical experiments

In this section, we consider two illustrative numerical example where we clearly show the performance of the optimal proposal pdf in the noisy IS setting (showing the variance gains in estimation, with respect to the use the optimal proposal density from the non-noisy setting). For simplicity, we consider one-dimensional scenarios, and test the optimal proposal pdf with different densities $p(x)$ (uniform and Gaussian), and different types of variance behavior, $\sigma(x)$.

First experiment. Let $p(x) = \frac{1}{b-a}$ for $x \in [a, b]$, i.e., a uniform density in $[a, b]$ with $a = 0.1$ and $b = 10$. We set $\widetilde{m}(\mathbf{x}) = p(\mathbf{x})e^{\epsilon}$ with $\epsilon \sim \mathcal{N}\left(-\sigma^2/2, \sigma^2\right)$ so that $\mathbb{E}[e^{\epsilon}] = 1$, and we have $m(x) = \mathbb{E}[\widetilde{m}(\mathbf{x})] = p(x)$.
We consider the estimation of $\bar{Z} = 1$ using the optimal proposal pdf $q_{\text{opt}}(x)$ in Eq. (7.11), and the optimal proposal pdf in the non-noisy setting, i.e., $q_{\text{sub-opt}}(x) = p(x)$. More specifically, we consider

$$\sigma(x) = A|\log(x)|, \quad A > 0.$$

Hence,

$$s(x)^2 = \frac{e^{\sigma(x)^2} - 1}{(b-a)^2}, \quad \text{and} \quad q_{\text{opt}}(x) \propto \frac{1}{b-a} e^{\sigma(x)^2}.$$

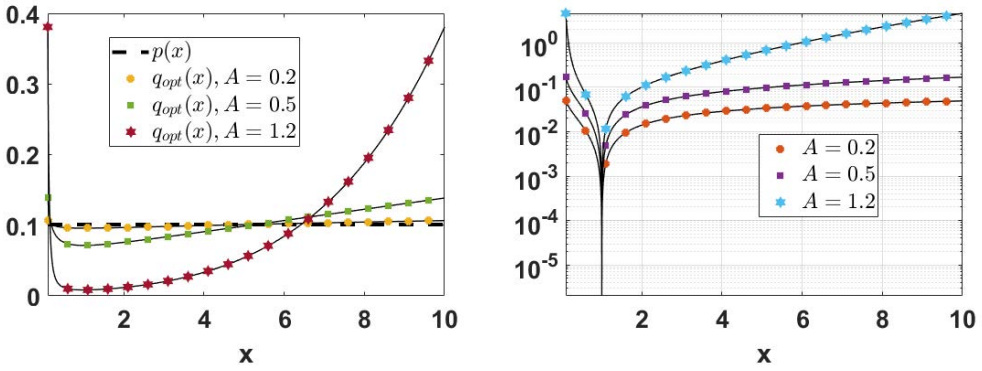
Clearly, by changing A, we change the form of both $s(x)^2$ and $q_{\text{opt}}(x)$. For instance, increasing A also increases the magnitude of $s(x)^2$ and hence the mismatch between $q_{\text{sub-opt}}(x) = p(x)$ and $q_{\text{opt}}(x)$, as depicted in Figure 7.1. Indeed, for $A = 0.2$, $q_{\text{opt}}(x)$ is almost identical to $p(x)$ since the magnitude of $s(x)$ is small w.r.t. the values of $p(x)$. As A increases, $q_{\text{opt}}(x)$ deviates from $p(x)$, being in the middle between $p(x)$ and $s(x)$, and eventually would converge to $s(x)$ for $A \gg 1$. It is also interesting to note that $q_{\text{opt}}(x)$ with $A = 1.2$ has very little probability mass around $x = 1$, where the noise is zero, since it needs to concentrate probability mass in the extremes of the interval, where the noise power is huge.

Let also denote as $V_{\text{sub-opt}}$ the variance obtained using $q_{\text{sub-opt}}(x) = p(x)$ given in Eq. (7.12), and $V_{\text{opt}} = V_{\text{min}}$ the variance obtained using $q_{\text{opt}}(x)$ given in Eq. (7.13). In Figure 7.3(a), we show the ratio of variances $\frac{V_{\text{sub-opt}}}{V_{\text{opt}}}$ both theoretically and empirically, as a function of A, where $V_{\text{sub-opt}}$ and V_{opt} are the variances of $\widehat{Z}$ when using $p(x)$ and $q_{\text{opt}}(x)$ as proposals, respectively. We can observe the clear advantage of using the optimal proposal density $q_{\text{opt}}(x)$ in Eq. (7.11).

Second experiment. Let us consider now a Gaussian pdf, $p(x) = \mathcal{N}(x|0, 1)$, and the same error model as in the previous example but considering

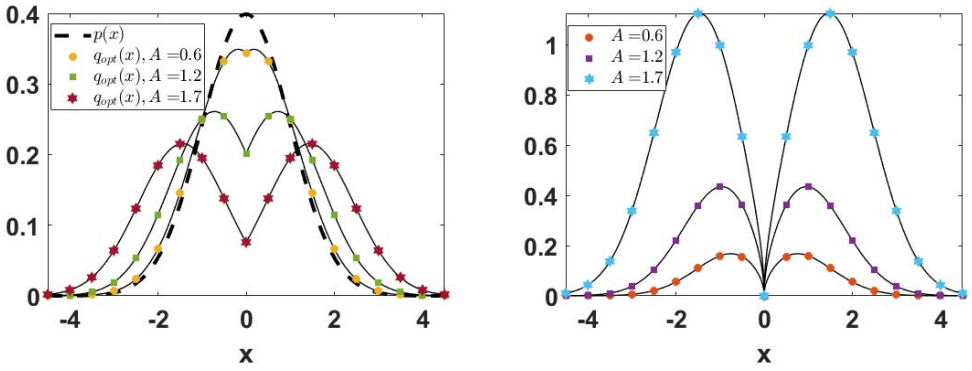
$$\sigma(x) = A|x|^{\frac{1}{2}}, \quad A > 0.$$

Figure 7.2 depicts the $q_{\text{opt}}(x)$ and $s(x)$, as a function of x, for several values of A. Note that, in this example, increasing A makes $q_{\text{opt}}(x)$ become bimodal. As in the previous example, as A increases, the optimal proposal $q_{\text{opt}}(x)$ will converge to $s(x)$. The theoretical and empirical curves of the ratio of variances, $\frac{V_{\text{sub-opt}}}{V_{\text{opt}}}$, in estimating Z are shown in Figure 7.3(b).



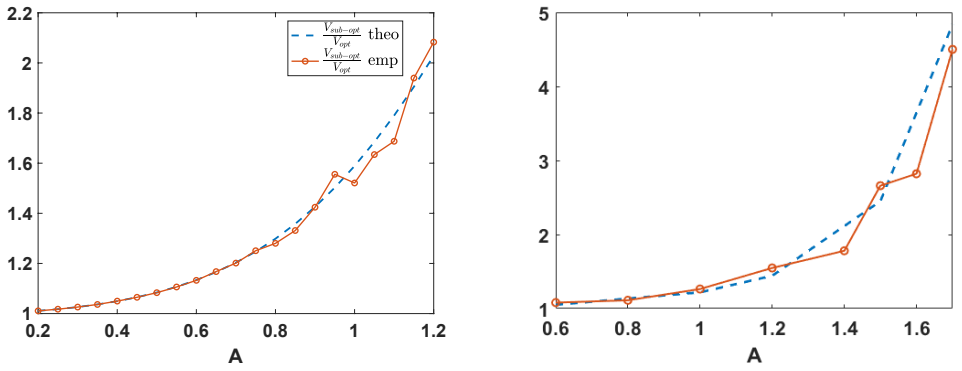
(a) $q_{\text{opt}}(x)$ for different values of A. (b) $s(x)$ for different values of A in log-scale.

Figure 7.1: Uniform example. **(a)** Optimal proposals $q_{\text{opt}}(x)$ for different values of A, and the $q_{\text{sub-opt}}(x) = p(x)$ in dashed line; **(b)** The standard deviation $s(x)$ for different values of A.



(a) $q_{\text{opt}}(x)$ for different values of A. (b) $s(x)$ for different values of A.

Figure 7.2: Gaussian example. **(a)** Optimal proposals $q_{\text{opt}}(x)$ for different values of A, and the $q_{\text{sub-opt}}(x) = p(x)$ in dashed line; **(b)** The standard deviation $s(x)^2$ for different values of A.



(a) Uniform example. (b) Gaussian example.

Figure 7.3: Theoretical and empirical ratio of variances $\frac{V_{\text{sub-opt}}}{V_{\text{opt}}}$ in the estimation of $\bar{Z} = 1$ for both experiments. The x-axis denotes the noise level (larger A means greater noise).

7.6. Conclusions

Working with noisy evaluations of the target density is usual in Monte Carlo, especially in the last years. In this work, we have analyzed the use of optimal proposal densities in a noisy IS framework. Previous works have focused on the trade-off between accuracy in the evaluation and computational cost in order to form optimal estimators. In this work, we have considered a general setting and derived the optimal proposals for the noisy IS estimators. These optimal proposals incorporate the variance function of the noisy evaluation in order to propose samples in regions that are more affected by noise. In this sense, we can informally state that the optimal proposal densities play the role of an acquisition function that also take into account the noise power.

7.7. Appendix

7.7.1. Covariance between $\widehat{E}_p$ and $\widehat{Z}$

We show that

$$\mathrm{Cov}[\widehat{E}_p, \widehat{Z}] = \frac{1}{N}\mathbb{E}\left[\frac{f_p(\mathbf{x})(m(\mathbf{x})^2 + s(\mathbf{x})^2)}{q(\mathbf{x})^2}\right] - \frac{1}{N}E_p\bar{Z}.$$

First, recall that $\mathrm{Cov}[\widehat{E}_p, \widehat{Z}] = \mathbb{E}[\widehat{E}_p\widehat{Z}] - E_p\bar{Z}$. By the law of iterated expectations,

$$\mathbb{E}[\widehat{E}_p\widehat{Z}] = \mathbb{E}[\mathbb{E}[\widehat{E}_p\widehat{Z}|\mathbf{x}_{1:N}]].$$

The inner expectation is

$$\begin{aligned}\mathbb{E}[\widehat{E}_p\widehat{Z}|\mathbf{x}_{1:N}] &= \mathbb{E}\left[\frac{1}{N^2}\sum_{i=1}^{N} w_i^2 f_p(\mathbf{x}_i) + \frac{2}{N^2}\sum_{i=1}^{N}\sum_{j>i}^{N} w_i w_j f_p(\mathbf{x}_i)\middle|\mathbf{x}_{1:N}\right] \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\frac{f_p(\mathbf{x}_i)(s(\mathbf{x}_i)^2 + m(\mathbf{x}_i)^2)}{q(\mathbf{x}_i)^2} + \frac{2}{N^2}\sum_{i=1}^{N}\sum_{j>i}^{N}\frac{m(\mathbf{x}_i)f(\mathbf{x}_i)}{q(\mathbf{x}_i)}\frac{m(\mathbf{x}_j)}{q(\mathbf{x}_j)}.\end{aligned}$$

Hence, we obtain

$$\begin{aligned}\mathbb{E}\left[\mathbb{E}[\widehat{E}_p\widehat{Z}|\mathbf{x}_{1:N}]\right] &= \frac{1}{N}\mathbb{E}\left[\frac{f(\mathbf{x})(s(\mathbf{x})^2 + m(\mathbf{x})^2)}{q(\mathbf{x})^2}\right] \\ &\quad + \frac{2}{N^2}\sum_{i=1}^{N}\sum_{j>i}^{N}\mathbb{E}\left[\frac{m(\mathbf{x}_i)f(\mathbf{x}_i)}{q(\mathbf{x}_i)}\right]\mathbb{E}\left[\frac{m(\mathbf{x}_j)}{q(\mathbf{x}_j)}\right] \\ &= \frac{1}{N}\mathbb{E}\left[\frac{f(\mathbf{x})(s(\mathbf{x})^2 + m(\mathbf{x})^2)}{q(\mathbf{x})^2}\right] + \frac{2}{N^2}\sum_{i=1}^{N}\sum_{j>i}^{N}E_p\bar{Z} \\ &= \frac{1}{N}\mathbb{E}\left[\frac{f(\mathbf{x})(s(\mathbf{x})^2 + m(\mathbf{x})^2)}{q(\mathbf{x})^2}\right] + E_p\bar{Z}\left(1 - \frac{1}{N}\right).\end{aligned}$$

Combining the results, we obtain the desired expression.

Bibliography

[1] L. Acerbi. Variational Bayesian Monte Carlo with noisy likelihoods. *arXiv:2006.08655*, 2020.

[2] O. D. Akyildiz, I. P. Marino, and J. Míguez. Adaptive noisy importance sampling for stochastic optimization. In *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 1–5. IEEE, 2017.

[3] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[4] C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

[5] D. V. Arnold. *Noisy optimization with evolution strategies*, volume 8. Springer Science & Business Media, 2012.

[6] R. Bardenet, A. Doucet, and C. Holmes. On Markov chain Monte Carlo methods for tall data. *The Journal of Machine Learning Research*, 18(1):1515–1557, 2017.

[7] N. Bliznyuk, D. Ruppert, C. Shoemaker, R. Regis, S. Wild, and P. Mugunthan. Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. *Journal of Computational and Graphical Statistics*, 17(2):270–294, 2008.

[8] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, and J.-B. Mouret. A survey on policy search algorithms for learning robot controllers in a handful of trials. *IEEE Transactions on Robotics*, 36(2):328–347, 2019.

[9] N. Chopin, P.E. Jacob, and O. Papaspiliopoulos. SMC2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426, 2013.

[10] D. Crisan and J. Miguez. Nested particle filters for online parameter estimation in discrete-time state-space markov models. *Bernoulli*, 24(4A):3039–3086, 2018.

[11] M. P. Deisenroth, G. Neumann, and J. Peters. A survey on policy search for robotics. *Foundations and trends in Robotics*, 2(1-2):388–403, 2013.

[12] A. Doucet, M. K Pitt, G. Deligiannidis, and R. Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 102(2):295–313, 2015.

[13] C. C. Drovandi, M. T. Moores, and R. J. Boys. Accelerating pseudo-marginal MCMC using Gaussian processes. *Computational Statistics & Data Analysis*, 118:1–17, 2018.

[14] A. B. Duncan, A. M. Stuart, and M.-T. Wolfram. Ensemble inference methods for models with noisy and expensive likelihoods. *arXiv:2104.03384*, 2021.

[15] P. Fearnhead, O. Papaspiliopoulos, and G. O. Roberts. Particle filters for partially observed diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):755–777, 2008.

[16] P. Fearnhead, O. Papaspiliopoulos, G. O. Roberts, and A. Stuart. Random-weight particle filtering of continuous time processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):497–512, 2010.

[17] M. U. Gutmann and J. Corander. Bayesian optimization for likelihood–free inference of simulator–based statistical models. *Journal of Machine Learning Research*, 2016.

[18] M. Järvenpää, M. U. Gutmann, A. Vehtari, and P. Marttinen. Parallel Gaussian process surrogate Bayesian inference with noisy likelihood evaluations. *Bayesian Analysis*, 16(1):147–178, 2021.

[19] F. Llorente, L. Martino, D. Delgado, and J. Lopez-Santiago. Marginal likelihood computation for model selection and hypothesis testing: an extensive review. *arXiv:2005.08334*, 2020.

[20] F. Llorente, L. Martino, V. Elvira, D. Delgado, and J. Lopez-Santiago. Adaptive quadrature schemes for Bayesian inference via active learning. *arXiv:2006.00535*, 2020.

[21] F. Llorente, L. Martino, J. Read, and D. Delgado. A survey of Monte Carlo methods for noisy and costly densities with application to reinforcement learning. *arXiv:2108.00490*, 2021.

[22] D. Luengo, L. Martino, M. Bugallo, V. Elvira, and S. Särkkä. A survey of Monte Carlo methods for parameter estimation. *EURASIP Journal on Advances in Signal Processing*, 2020:1–62, 2020.

[23] L. Martino, J. Read, V. Elvira, and F. Louzada. Cooperative parallel particle filters for on-line model selection and applications to urban mobility. *Digital Signal Processing*, 60:172–185, 2017.

[24] J. Park and M. Haran. Bayesian inference in the presence of intractable normalizing functions. *Journal of the American Statistical Association*, 113(523):1372–1390, 2018.

[25] L. F. Price, C. C. Drovandi, A. Lee, and D. J. Nott. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1):1–11, 2018.

[26] M. Quiroz, R. Kohn, M. Villani, and M.-N. Tran. Speeding up MCMC by efficient data subsampling. *Journal of the American Statistical Association*, 2018.

[27] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[28] A. Stuart and K. Ord. *Kendall's Advanced Theory of Statistics*. Arnold, London, 1998.

[29] D. H. Svendsen, L. Martino, and G. Camps-Valls. Active emulation of computer codes with Gaussian processes–Application to remote sensing. *Pattern Recognition*, 100:107103, 2020.

[30] M.-N. Tran, M. Scharth, M. K. Pitt, and R. Kohn. Importance sampling squared for Bayesian inference in latent variable models. *arXiv preprint arXiv:1309.3339*, 2013.

8. MARGINAL LIKELIHOOD COMPUTATION FOR MODEL SELECTION AND HYPOTHESIS TESTING: AN EXTENSIVE REVIEW

Extended version of:
In *SIAM Review (to appear)*, (2023)

F. Llorente⋆, L. Martino⋆⋆, D. Delgado⋆, J. Lopez-Santiago⋆
⋆ Universidad Carlos III de Madrid, Leganés (Spain).
⋆⋆ Universidad Rey Juan Carlos, Fuenlabrada (Spain).

Abstract

This is an up-to-date introduction to, and overview of, marginal likelihood computation for model selection and hypothesis testing. Computing normalizing constants of probability models (or ratio of constants) is a fundamental issue in many applications in statistics, applied mathematics, signal processing and machine learning. This article provides a comprehensive study of the state-of-the-art of the topic. We highlight limitations, benefits, connections and differences among the different techniques. Problems and possible solutions with the use of improper priors are also described. Some of the most relevant methodologies are compared through theoretical comparisons and numerical experiments.

Keywords: Marginal likelihood, Bayesian evidence, numerical integration, model selection, hypothesis testing, quadrature rules, double-intractable posteriors, partition functions.

8.1. Introduction

Marginal likelihood (a.k.a., Bayesian evidence) and Bayes factors are the core of the Bayesian theory for testing hypotheses and model selection [54, 88]. More generally, the computation of normalizing constants or ratios of normalizing constants has played an important role in statistical physics and numerical analysis [99]. In the Bayesian setting, the approximation of normalizing constants is also required in the study of the so-called double intractable posteriors [53].

Several methods have been proposed for approximating the marginal likelihood and normalizing constants in the last decades. Most of these techniques have been originally introduced in the field of statistical mechanics. Indeed, the marginal likelihood is the analogous of a central quantity in statistical physics known as the *partition function* which is

also closely related to another important quantity often called *free-energy*. The relationship between statistical physics and Bayesian inference has been remarked in different works [3, 51].

The model selection problem has been also addressed from different points of view. Several criteria have been proposed to deal with the trade-off between the goodness-of-fit of the model and its simplicity. For instance, the Akaike information criterion (AIC) or the focused information criterion (FIC) are two examples of these approaches [91, 19]. The Bayesian-Schwarz information criterion (BIC) is related to the marginal likelihood approximation, as discussed in Section 8.3. The deviance information criterion (DIC) is a generalization of the AIC, which is often used in Bayesian inference [97, 98]. It is particularly useful for hierarchical models and it can be approximately computed when the outputs of a Markov Chain Monte Carlo (MCMC) algorithm are given. However, DIC is not directly related to the Bayesian evidence [85]. Another different approach, also based on information theory, is the so-called minimum description length principle (MDL) [43]. MDL was originally derived for data compression, and then was applied to model selection and hypothesis testing. Roughly speaking, MDL considers that the best explanation for a given set of data is provided by the *shortest description* of that data [43].
In the Bayesian framework, there are two main classes of sampling algorithms. The first one consists in approximating the marginal likelihood of different models or the ratio of two marginal likelihoods. In this work, we focus on this first approach. The second sampling approach extends the posterior space including a discrete indicator variable m, denoting the m-th model [11, 42]. For instance, in the well-known **reversible jump MCMC** [42], a Markov chain is generated in this extended space, allowing jumps between models with possibly different dimensions. However, generally, these methods are difficult to tune and the mixing of the chain can be poor [45]. For further details, see also the interesting works [22, 41, 20]. The average number of MCMC iterations when the chain jumps or stays into the m-th model is proportional to the marginal likelihood of the corresponding model.
In this work, we provide an extensive review of computational techniques for the marginal likelihood computation. The main contribution is to present jointly numerous computational schemes (introduced independently in the literature) with a detailed description under the same notation, highlighting their differences, relationships, limitations and strengths. Most of them are based on the importance sampling (IS) approach and several of them are combination the MCMC and IS schemes. It is also important to remark that parts of the presented material are also novel, i.e., no contained in previous works. We have widely studied, analyzed and jointly described with a unique notation and classification, the methodologies presented in a vast literature from 1990s to the recent proposed algorithms (see Table 8.24). We also discuss issues and solutions when improper priors are employed. Therefore, this survey provides an ample covering of the literature, where we highlight important details and comparisons in order to facilitate the understanding of the interested readers and practitioners.
The problem statement and the main notation are introduced in the Section 8.2.1. Rele-

vant considerations regarding the marginal likelihood and other model selection strategies are given in Section 8.2.2 and Section 8.7. Specifically, a description of how the marginal likelihood handles the model fit and the model complexity is provided in Section 8.2.2. The dependence on the prior selection and the possible choice of an improper prior are discussed in Section 8.7. The different techniques have been classified in four main families, as shown in Section 8.2.3. Sections 8.3, 8.4, 8.5, 8.6 are devoted to the detailed description of the computational schemes for approximating the Bayesian evidence. Section 8.8 contains some numerical experiments. In Section 8.9, we conclude with a final summary and discussion. We provide also theoretical analyses of some of the experiments and other comparisons in the Supplementary Material.

8.2. Problem statement and preliminary discussions

8.2.1. Framework and notation

In many applications, the goal is to make inference about a variable of interest, $\boldsymbol{\theta} = \theta_{1:D_\theta} = [\theta_1, \theta_2, \ldots, \theta_{D_\theta}] \in \Theta \subseteq \mathbb{R}^{D_\theta}$, where $\theta_d \in \mathbb{R}$ for all $d = 1, \ldots, D_\theta$, given a set of observed measurements, $\mathbf{y} = [y_1, \ldots, y_{D_y}] \in \mathbb{R}^{D_y}$. In the Bayesian framework, one complete model $\mathcal{M}$ is formed by a likelihood function $\ell(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M})$ and a prior probability density function (pdf) $g(\boldsymbol{\theta}|\mathcal{M})$. All the statistical information is summarized by the posterior pdf, i.e.,

$$P(\boldsymbol{\theta}|\mathbf{y}, \mathcal{M}) = \frac{\ell(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M}) g(\boldsymbol{\theta}|\mathcal{M})}{p(\mathbf{y}|\mathcal{M})}, \tag{8.1}$$

where

$$Z = p(\mathbf{y}|\mathcal{M}) = \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M}) g(\boldsymbol{\theta}|\mathcal{M}) d\boldsymbol{\theta}, \tag{8.2}$$

is the so-called marginal likelihood, a.k.a., Bayesian evidence. This quantity is important for model selection purpose, as we show below. However, usually $Z = p(\mathbf{y}|\mathcal{M})$ is unknown and difficult to approximate, so that in many cases we are only able to evaluate the unnormalized target function,

$$\pi(\boldsymbol{\theta}|\mathbf{y}, \mathcal{M}) = \ell(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M}) g(\boldsymbol{\theta}|\mathcal{M}). \tag{8.3}$$

Note that $P(\boldsymbol{\theta}|\mathbf{y}, \mathcal{M}) \propto \pi(\boldsymbol{\theta}|\mathbf{y}, \mathcal{M})$ [54, 88]. For the sake of simplicity, hereafter we use the simplified notation $P(\boldsymbol{\theta}|\mathbf{y})$ and $\pi(\boldsymbol{\theta}|\mathbf{y})$. Thus, note that

$$Z = \int_\Theta \pi(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta}. \tag{8.4}$$

Model Selection and testing hypotheses. Let us consider now M possible models (or hypotheses), $\mathcal{M}_1, ..., \mathcal{M}_M$, with prior probability mass $p_m = \mathbb{P}(\mathcal{M}_m)$, $m = 1, ..., M$. Note that, we can have variables of interest $\boldsymbol{\theta}^{(m)} = [\theta_1^{(m)}, \theta_2^{(m)}, \ldots, \theta_{D_m}^{(m)}] \in \Theta_m \in \mathbb{R}^{D_m}$, with

possibly different dimensions in the different models. The posterior of the m-th model is given by

$$p(\mathcal{M}_m|\mathbf{y}) = \frac{p_m p(\mathbf{y}|\mathcal{M}_m)}{p(\mathbf{y})} \propto p_m Z_m \tag{8.5}$$

where $Z_m = p(\mathbf{y}|\mathcal{M}_m) = \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta}_m, \mathcal{M}_m) g(\boldsymbol{\theta}_m|\mathcal{M}_m) d\boldsymbol{\theta}_m$, and $p(\mathbf{y}) = \sum_{m=1}^{M} p(\mathcal{M}_m) p(\mathbf{y}|\mathcal{M}_m)$. Moreover, the ratio of two marginal likelihoods

$$\frac{Z_m}{Z_{m'}} = \frac{p(\mathbf{y}|\mathcal{M}_m)}{p(\mathbf{y}|\mathcal{M}_{m'})} = \frac{p(\mathcal{M}_m|\mathbf{y})/p_m}{p(\mathcal{M}_{m'}|\mathbf{y})/p_{m'}}, \tag{8.6}$$

also known as *Bayes factors*, represents the posterior to prior odds of models m and m'. If some quantity of interest is common to all models, the posterior of this quantity can be studied via *model averaging* [46], i.e., a complete posterior distribution as a mixture of M partial posteriors linearly combined with weights proportionally to $p(\mathcal{M}_m|\mathbf{y})$ (see, e..g, [72, 101]). Therefore, in all these scenarios, we need the computation of Z_m for all $m = 1, ..., M$. In this work, we describe different computational techniques for calculating Z_m, mostly based on Markov Chain Monte Carlo (MCMC) and Importance Sampling (IS) algorithms [88]. Hereafter, we assume proper prior $g(\boldsymbol{\theta}|\mathcal{M}_m)$. Regarding the use of *improper priors* see Section 8.7.2. Moreover, we usually denote Z, Θ, $\mathcal{M}$, omitting the subindex m, to simplify notation. It is important also to remark that, in some cases, it is also necessary to approximate normalizing constants (that are also functions of the parameters) in each iteration of an MCMC algorithm, in order to allow the study of the posterior density. For instance, this is the case of the so-called double intractable posteriors [53].

Remark 1. *The evidence Z is the normalizing constant of $\pi(\boldsymbol{\theta}|\mathbf{y})$, hence most of the methods in this review can be used to approximate normalizing constants of generic pdfs.*

Remark 2. *Instead of approximating the single values Z_m for all m, another approach consists in estimating directly the ratio of two marginal likelihoods $\frac{Z_m}{Z_{m'}}$, i.e., approximating directly the Bayes factors. For these reasons, several computational methods focus on estimating the ratio of two normalizing constants. However, they can be used also for estimating a single Z_m provided that $Z_{m'}$ is known.*

Table 8.1: Main notation of the work.

D_θ	dimension of the parameter space, $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^{D_\theta}$.
D_y	Total number of data.
$\boldsymbol{\theta}$	parameters; $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_{D_\theta}]$.
$\mathbf{y}$	Data, $\mathbf{y} = [y_1, \ldots, y_{D_y}]$.
$\ell(\mathbf{y}\|\boldsymbol{\theta})$	Likelihood function.
$g(\boldsymbol{\theta})$	Prior pdf.
$P(\boldsymbol{\theta}\|\mathbf{y})$	Posterior pdf, $P(\boldsymbol{\theta}\|\mathbf{y}) = \frac{\ell(\mathbf{y}\|\boldsymbol{\theta})g(\boldsymbol{\theta})}{Z}$.
$\pi(\boldsymbol{\theta}\|\mathbf{y})$	Unnormalized posterior, $\pi(\boldsymbol{\theta}\|\mathbf{y}) = \ell(\mathbf{y}\|\boldsymbol{\theta})g(\boldsymbol{\theta}) \propto P(\boldsymbol{\theta}\|\mathbf{y})$.
$Z = p(\mathbf{y})$	Marginal likelihood, a.k.a., Bayesian evidence $Z = \int_\Theta \pi(\boldsymbol{\theta}\|\mathbf{y})d\boldsymbol{\theta}$.
$\bar{q}(\boldsymbol{\theta})$	Proposal pdf.
$q(\boldsymbol{\theta})$	Unnormalized proposal function, $q(\boldsymbol{\theta}) \propto \bar{q}(\boldsymbol{\theta})$.

8.2.2. Model fit and model complexity

Bounds of the evidence Z

Let us denote the maximum and minimum value of the likelihood function as $\ell_{\min} = \ell(\mathbf{y}|\boldsymbol{\theta}_{\min}) = \min_{\boldsymbol{\theta}\in\theta} \ell(\mathbf{y}|\boldsymbol{\theta})$, and $\ell_{\max} = \ell(\mathbf{y}|\boldsymbol{\theta}_{\max}) = \max_{\boldsymbol{\theta}\in\theta} \ell(\mathbf{y}|\boldsymbol{\theta})$, respectively. Note that

$$Z = \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})d\boldsymbol{\theta} \leq \ell(\mathbf{y}|\boldsymbol{\theta}_{\max}) \int_\Theta g(\boldsymbol{\theta})d\boldsymbol{\theta} = \ell(\mathbf{y}|\boldsymbol{\theta}_{\max}).$$

Similarly, we can obtain $Z \geq \ell(\mathbf{y}|\boldsymbol{\theta}_{\min})$. The maximum and minimum value of Z are reached with two degenerate choices of the prior, $g(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{\max})$ and $g(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{\min})$. Hence, for every other choice of $g(\boldsymbol{\theta})$, we have

$$\ell(\mathbf{y}|\boldsymbol{\theta}_{\min}) \leq Z \leq \ell(\mathbf{y}|\boldsymbol{\theta}_{\max}). \tag{8.7}$$

Namely, depending on the choice of the prior $g(\boldsymbol{\theta})$, we can have any value of Bayesian evidence contained in the interval $[\ell(\mathbf{y}|\boldsymbol{\theta}_{\min}), \ell(\mathbf{y}|\boldsymbol{\theta}_{\max})]$. For further discussion see Section 8.7.

The two possible extreme values correspond to the worst and the best model fit, respectively. Below, we will see that if $Z = \ell(\mathbf{y}|\boldsymbol{\theta}_{\min})$ the chosen prior, $g(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{\min})$, applies the greatest possible penalty to the model whereas, if $Z = \ell(\mathbf{y}|\boldsymbol{\theta}_{\max})$, the chosen prior, $g(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{\max})$, does not apply any penalization to the model complexity (we have the maximum overfitting). Namely, the evidence Z is an average of the likelihood values, weighted according to the prior.

Occam factor and implicit/intrinsic complexity penalization in Z

The marginal likelihood can be expressed as

$$Z = \ell_{\max} W, \tag{8.8}$$

where $W \in [0, 1]$ is the *Occam factor* [48, Sect. 3]. More specifically, the Occam factor is defined as

$$W = \frac{1}{\ell_{\max}} \int_{\Theta} g(\boldsymbol{\theta}) \ell(\mathbf{y}|\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{8.9}$$

and it is $\frac{\ell_{\min}}{\ell_{\max}} \leq W \leq 1$. The factor W measures the penalty of the model complexity *intrinsically* contained in the marginal likelihood Z: this penalization depends on the chosen prior and the number of data involved. We show below that the Occam factor measures the "overlap" between likelihood and prior, i.e., how diffuse the prior is with respect to the likelihood function. Finally, it is important to remark that, considering the posterior of the m-th model $p(\mathcal{M}_m|\mathbf{y})$, we have another possible penalization term due to the prior $p_m = P(\mathcal{M}_m) \in [0, 1]$, i.e.,

$$p(\mathcal{M}_m|\mathbf{y}) \propto Z p_m = \ell_{\max} W p_m = \ell_{\max} \widetilde{W},$$

where we have defined the *posterior Occam factor* as $\widetilde{W} = W p_m$.

Occam factor with uniform priors

One-dimensional case. Let start with a single parameter, $\boldsymbol{\theta} = \theta$, and a uniform prior in $[a, b]$. We can define the amount of the likelihood mass is contained inside the prior bounds,

$$\Delta_\ell = \frac{1}{\ell_{\max}} \int_a^b \ell(\mathbf{y}|\theta) d\theta, \quad \text{where} \quad \ell_{\max} = \max_{\theta \in \theta} \ell(\mathbf{y}|\theta). \tag{8.10}$$

Defining also the width of the prior as $\Delta_\theta = |\Theta| = b - a$, note that $0 \leq \Delta_\ell \leq \Delta_\theta$, where the equality $\Delta_\ell = \Delta_\theta$ is given when the likelihood is $\ell(\mathbf{y}|\theta) = \ell_{\max}$ is constant. The Occam factor is given as the ratio of Δ_ℓ and the width of a uniform prior Δ_θ [48],

$$W = \frac{\Delta_\ell}{\Delta_\theta}. \tag{8.11}$$

If the likelihood function is integrable in $\mathbb{R}$, then there exists a finite upper bound for Δ_ℓ when $\Delta_\theta \to \infty$, that is $\Delta_\ell^* = \frac{1}{\ell_{\max}} \int_{-\infty}^{+\infty} \ell(\mathbf{y}|\theta) d\theta$. Hence, in this scenario, we can see that an increase of Δ_θ makes that W approaches 0.

Multidimensional case. Consider now a multidimensional case, $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_{D_\theta}] \in \Theta \subseteq \mathbb{R}^{D_\theta}$, where we can use the same uniform prior, with the same width $\Delta_\theta = |\Theta|$, for

all the parameters. In this case, $\Delta_\ell = \frac{1}{\ell_{\max}} \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta}) d\boldsymbol{\theta} \leq (\Delta_\theta)^{D_\theta}$ is D_θ-dimensional integral, and $\ell_{\max} = \max \ell(\mathbf{y}|\boldsymbol{\theta})$. Then, for D_θ parameters, the Occam factor is

$$W = \frac{\Delta_\ell}{(\Delta_\theta)^{D_\theta}}. \tag{8.12}$$

Usually, as D_θ grows, the fitting improves until reaching (or approaching) a maximum, possible overfitting. Then, with D_θ big enough, $\ell_{\max}$ tends to be virtually constant (reaching the maximum overfitting). If $\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta}) d\boldsymbol{\theta}$ grows slower than $(\Delta_\theta)^{D_\theta}$ as $D_\theta \to \infty$, and assuming for an illustrative purpose $\Delta_\theta > 1$, then W converges to 0 as $D_\theta \to \infty$. That is, when we introduce more and more parameters, the increase in model fit will be dominated, at some point, by the model complexity penalization implicitly contained in the evidence Z.

Marginal likelihood and information criteria

Considering the expressions (8.8) and (8.12) and taking the logarithm, we obtain

$$\begin{aligned} \log Z = \log \ell_{\max} + \log W &= \log \ell_{\max} + \log \Delta_\ell - D_\theta \log \Delta_\theta, \\ &= \log \ell_{\max} + \eta D_\theta, \end{aligned} \tag{8.13}$$

where $\eta = \frac{\log \Delta_\ell}{D_\theta} - \log \Delta_\theta$ is a constant value, which also depends on the number of data D_y and, generally, $\eta = \eta(D_y, D_\theta)$. Different model selection rules in the literature consider the simplification $\eta = \eta(D_y)$. Note that $\log \ell_{\max}$ is a fitting term whereas ηD_θ is a penalty for the model complexity. Instead of maximizing Z (or $\log Z$) for model selection purposes, several authors consider the minimization of some cost functions derived by different information criteria. To connect them with the marginal likelihood maximization, we consider the expression of $-2 \log Z = -2I$ where $I = -\log Z$ resembles the Shannon information associated to $Z = p(\mathbf{y})$, i.e.,

$$2I = -2 \log Z = -2 \log \ell_{\max} - 2\eta D_\theta. \tag{8.14}$$

The expression above encompasses several well-known information criteria proposed in the literature and shown in Table 8.2, which differ for the choice of η. In all these cases, η is just a function of the number of data D_y. More details regarding these information criteria are given in Section 8.3.

Remark 3. *The penalty term in the information criteria is the same for every parameter. The Bayesian approach allows the choice of different penalties, assuming different priors, one for each parameter.*

8.2.3. A general overview of the computational methods

After a depth revision of the literature, we have recognized four main families of techniques, described below. We list them in order of complexity, from the simplest to the

Table 8.2: Different information criterion for model selection.

Criterion	Choice - approximation of η
Bayesian-Schwarz information criterion (BIC) [94]	$-\frac{1}{2}\log D_y$
Akaike information criterion (AIC) [97]	-1
Hannan-Quinn information criterion (HQIC) [44]	$-\log(\log(D_y))$

most complex underlying main idea. However, each class can contain both simple and very sophisticated algorithms.

Family 1: *Deterministic approximations***.** These methods consider an analytical approximation of the function $P(\boldsymbol{\theta}|\mathbf{y})$. The Laplace method and the Bayesian Information Criterion (BIC), belongs to this family (see Section 8.3).

Family 2: *Methods based on density estimation***.** This class of algorithms uses the equality

$$\widehat{Z} = \frac{\pi(\boldsymbol{\theta}^*|\mathbf{y})}{\widehat{P}(\boldsymbol{\theta}^*|\mathbf{y})}, \tag{8.15}$$

where $\widehat{P}(\boldsymbol{\theta}^*|\mathbf{y}) \approx P(\boldsymbol{\theta}^*|\mathbf{y})$ represents an estimation of the density $P(\boldsymbol{\theta}|\mathbf{y})$ at some point $\boldsymbol{\theta}^*$. Generally, the point $\boldsymbol{\theta}^*$ is chosen in a high-probability region. The techniques in this family differ in the procedure employed for obtaining the estimation $\widehat{P}(\boldsymbol{\theta}^*|\mathbf{y})$. One famous example is the Chib's method [15]. Section 8.3 is devoted to describe methods belonging to family 1 and family 2.

Family 3: *Importance sampling (IS) schemes.* The IS methods are based on rewriting Eq. (8.2) as an expected value w.r.t. a simpler normalized density $\bar{q}(\boldsymbol{\theta})$, i.e., $Z = \int_\Theta \pi(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} = E_{\bar{q}}\left[\frac{\pi(\boldsymbol{\theta}|\mathbf{y})}{\bar{q}(\boldsymbol{\theta})}\right]$. This is the most considered class of methods in the literature, containing numerous variants, extensions and generalizations. We devote Sections 8.4-8.5 to this family of techniques.

Family 4: *Methods based on a vertical representation.* These schemes rely on changing the expression of $Z = \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})d\boldsymbol{\theta}$ (that is a multidimensional integral) to equivalent one-dimensional integrals [84, 105, 95]. Then, a quadrature scheme is applied to approximate this one-dimensional integral. The most famous example is the nested sampling algorithm [95]. Section 8.6 is devoted to this class of methods.

8.3. Methods based on deterministic approximations and density estimation

In this section, we consider approximations of $P(\boldsymbol{\theta}|\mathbf{y})$, or its unnormalized version $\pi(\boldsymbol{\theta}|\mathbf{y})$, in order to obtain an estimation Z. In a first approach, the methods consider $P(\boldsymbol{\theta}|\mathbf{y})$ or $\pi(\boldsymbol{\theta}|\mathbf{y})$ as a function, and try to obtain a good approximation given another parametric or

non-parametric family of functions. Another approach consists in approximating $P(\boldsymbol{\theta}|\mathbf{y})$ only at one specific point $\boldsymbol{\theta}^*$, i.e., $\widehat{P}(\boldsymbol{\theta}^*|\mathbf{y}) \approx P(\boldsymbol{\theta}^*|\mathbf{y})$ ($\boldsymbol{\theta}^*$ is usually chosen in high posterior probability regions), and then using the identity

$$\widehat{Z} = \frac{\pi(\boldsymbol{\theta}^*|\mathbf{y})}{\widehat{P}(\boldsymbol{\theta}^*|\mathbf{y})}. \tag{8.16}$$

The latter scheme is often called *candidate's estimation*.

8.3.1. Laplace's method

Let us define $\widehat{\boldsymbol{\theta}}_{\text{MAP}} \approx \boldsymbol{\theta}_{\text{MAP}} = \arg\max P(\boldsymbol{\theta}|\mathbf{y})$ (obtained by some optimization method), which is an approximation of the *maximum a posteriori* (MAP), and consider a Gaussian approximation of $P(\boldsymbol{\theta}|\mathbf{y})$ around $\widehat{\boldsymbol{\theta}}_{\text{MAP}}$, i.e.,

$$\widehat{P}(\boldsymbol{\theta}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \widehat{\boldsymbol{\Sigma}}), \tag{8.17}$$

with $\widehat{\boldsymbol{\Sigma}} \approx -\mathbf{H}^{-1}$, which is an approximation of the negative inverse Hessian matrix of $\log \pi(\boldsymbol{\theta}|\mathbf{y})$ at $\widehat{\boldsymbol{\theta}}_{\text{MAP}}$. Replacing in Eq. (8.16), with $\boldsymbol{\theta}^* = \widehat{\boldsymbol{\theta}}_{\text{MAP}}$, we obtain the Laplace approximation

$$\widehat{Z} = \frac{\pi(\widehat{\boldsymbol{\theta}}_{\text{MAP}}|\mathbf{y})}{\mathcal{N}(\widehat{\boldsymbol{\theta}}_{\text{MAP}}|\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \widehat{\boldsymbol{\Sigma}})} = (2\pi)^{\frac{D_x}{2}} |\widehat{\boldsymbol{\Sigma}}|^{\frac{1}{2}} \pi(\widehat{\boldsymbol{\theta}}_{\text{MAP}|\mathbf{y}}). \tag{8.18}$$

This is equivalent to the classical derivation of Laplace's estimator, which is based on expanding the $\log \pi(\boldsymbol{\theta}|\mathbf{y}) = \log(\ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta}))$ as quadratic around $\widehat{\boldsymbol{\theta}}_{\text{MAP}}$ and substituting in $Z = \int \pi(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}$, that is,

$$Z = \int \pi(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} = \int \exp\{\log \pi(\boldsymbol{\theta}|\mathbf{y})\}d\boldsymbol{\theta} \tag{8.19}$$

$$\approx \int \exp\left\{\log \pi(\widehat{\boldsymbol{\theta}}_{\text{MAP}}|\mathbf{y}) - \frac{1}{2}(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{\text{MAP}})^T \widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{\text{MAP}})\right\} d\boldsymbol{\theta} \tag{8.20}$$

$$= (2\pi)^{\frac{D_x}{2}} |\widehat{\boldsymbol{\Sigma}}|^{\frac{1}{2}} \pi(\widehat{\boldsymbol{\theta}}_{\text{MAP}}|\mathbf{y}). \tag{8.21}$$

In [52], they propose to use samples generated by a Metropolis-Hastings algorithm to estimate the quantities $\widehat{\boldsymbol{\theta}}_{\text{MAP}}$ and $\widehat{\boldsymbol{\Sigma}}$ [88]. The resulting method is called Laplace-Metropolis estimator. The authors in [23] present different variants of the Laplace's estimator. A relevant extension for Gaussian Markov random field models, is the so-called *integrated nested Laplace approximation* (INLA) [90].

8.3.2. Bayesian-Schwarz information criterion (BIC)

Let us define $\widehat{\boldsymbol{\theta}}_{\text{MLE}} \approx \boldsymbol{\theta}_{\text{MLE}} = \arg\max \ell(\mathbf{y}|\boldsymbol{\theta})$. The following quantity

$$\text{BIC} = D_\theta \log D_y - 2 \log \ell(\mathbf{y}|\widehat{\boldsymbol{\theta}}_{\text{MLE}}), \tag{8.22}$$

was introduced by Gideon E. Schwarz in [94], where D_θ represents the number of parameters of the model ($\boldsymbol{\theta} \in \mathbb{R}^{D_\theta}$), D_y is the number of data,[13] and $\ell(\mathbf{y}|\widehat{\boldsymbol{\theta}}_{\text{MLE}})$ is the estimated maximum value of the likelihood function. The value of $\widehat{\boldsymbol{\theta}}_{\text{MLE}}$ can be obtained using samples generated by a MCMC scheme. The BIC expression can be derived similarly to the Laplace's method, but this time with a second-order Taylor expansion of the $\log Z$ around its maximum $\boldsymbol{\theta}_{\text{MLE}}$ and a first-order expansion of the prior around $\boldsymbol{\theta}_{\text{MLE}}$ [50, Ch. 9.1.3]. The derivation is given in the Supplementary Material. Then, the final approximation is

$$Z \approx \widehat{Z} = \exp\left(\log \ell(\mathbf{y}|\widehat{\boldsymbol{\theta}}_{\text{MLE}}) - \frac{D_\theta}{2}\log D_y\right) = \exp\left(-\frac{1}{2}\text{BIC}\right), \quad \text{as } D_y \to \infty, \tag{8.23}$$

and $\text{BIC} \approx -2\log Z$, asymptotically as the number of data D_y grows. Then, smaller BIC values are associated to better models. Note that BIC clearly takes into account the complexity of the model since higher BIC values are given to models with more number of parameters D_θ. Namely the penalty $D_\theta \log D_y$ discourages overfitting, since increasing the number of parameters generally improves the goodness of the fit. Other criteria can be found in the literature, such as the well-known Akaike information criterion (AIC),

$$\text{AIC} = 2D_\theta - 2\log \ell(\mathbf{y}|\widehat{\boldsymbol{\theta}}_{\text{MLE}}).$$

However, they are not an approximation of the marginal likelihood Z and are usually founded on information theory derivations. Generally, they have the form of $c_p - 2\log \ell(\mathbf{y}|\widehat{\boldsymbol{\theta}}_{\text{MLE}})$ where the penalty term c_p of the model complexity changes in each different criterion (e.g., $c_p = D_\theta \log D_y$ in BIC and $c_p = 2D_\theta$ in AIC). Another example that uses MCMC samples is the Deviance Information Criterion (DIC), i.e.,

$$\text{DIC} = -\frac{4}{N}\sum_{n=1}^{N}\log \ell(\mathbf{y}|\boldsymbol{\theta}_n) - 2\log \ell(\mathbf{y}|\bar{\boldsymbol{\theta}}), \quad \text{where} \quad \bar{\boldsymbol{\theta}} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\theta}_n, \tag{8.24}$$

and $\{\boldsymbol{\theta}_n\}_{n=1}^N$ are outputs of an MCMC algorithm [97]. In this case, note that $c_p = -\frac{4}{N}\sum_{i=1}^{N}\log \ell(\mathbf{y}|\boldsymbol{\theta}_n)$. DIC is considered more adequate for hierarchical models than AIC, BIC [97], but is not directly related to the marginal likelihood [85]. See also related comments in Section 8.2.2.

8.3.3. Kernel density estimation (KDE)

KDE can be used to approximate the value of the posterior density at a given point $\boldsymbol{\theta}^*$, and then consider $Z \approx \frac{\pi(\boldsymbol{\theta}^*|\mathbf{y})}{\widehat{P}(\boldsymbol{\theta}^*|\mathbf{y})}$. For instance, we can build a kernel density estimate (KDE) of $P(\boldsymbol{\theta}|\mathbf{y})$ based on M samples distributed according to the posterior (obtained via an MCMC algorithm, for instance) by using M normalized kernel functions $k(\boldsymbol{\theta}|\boldsymbol{\mu}_m, h)$ (with $\int_\Theta k(\boldsymbol{\theta}|\boldsymbol{\mu}_m, h)d\boldsymbol{\theta} = 1$ for all m) where $\boldsymbol{\mu}_m$ is a location parameter and h is a scale parameter,

$$\widehat{P}(\boldsymbol{\theta}^*|\mathbf{y}) = \frac{1}{M}\sum_{m=1}^{M}k(\boldsymbol{\theta}^*|\boldsymbol{\mu}_m, h), \quad \{\boldsymbol{\mu}_m\}_{m=1}^M \sim P(\boldsymbol{\theta}|\mathbf{y}) \quad \text{(e.g., via MCMC)}. \tag{8.25}$$

[13] Note that, for simplicity, we are considering scalar observations y_i, so that the dimension D_y of the data vector $\mathbf{y}$ coincides with the number of data.

Generally, $\widehat{P}(\boldsymbol{\theta}^*|\mathbf{y})$ is a biased estimation of $P(\boldsymbol{\theta}^*|\mathbf{y})$. The estimator is $\widehat{Z} = \frac{\pi(\boldsymbol{\theta}^*|\mathbf{y})}{\widehat{P}(\boldsymbol{\theta}^*|\mathbf{y})}$ where the point $\boldsymbol{\theta}^*$ can be chosen as $\widehat{\boldsymbol{\theta}}_{\text{MAP}}$. If we consider N different points $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N$ (selected without any specific rule) we can also write a more general approximation,

$$\widehat{Z} = \frac{1}{N} \sum_{n=1}^{N} \frac{\pi(\boldsymbol{\theta}_n|\mathbf{y})}{\widehat{P}(\boldsymbol{\theta}_n|\mathbf{y})}. \tag{8.26}$$

Remark 4. *The estimator above is generally biased and depends on the choices of* **(a)** *of the points* $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N$, **(b)** *the scale parameter* h, *and* **(c)** *the number of samples* M *for building* $\widehat{P}(\boldsymbol{\theta}^*|\mathbf{y})$.

Remark 5. *A improved version of this approximation can be obtained by the importance sampling approach described in Sect. 8.4, where* $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N$ *are drawn from the KDE mixture* $\widehat{P}(\boldsymbol{\theta}|\mathbf{y})$. *In this case, the resulting estimator is unbiased.*

8.3.4. Chib's method

In [15, 16], the authors present more sophisticated methods to estimate $P(\boldsymbol{\theta}^*|\mathbf{y})$ using outputs from Gibbs sampling and the Metropolis-Hastings (MH) algorithm respectively [88]. Here we only present the latter method, since it can be applied in more general settings. In [16], the authors propose to estimate the value of the posterior at one point $\boldsymbol{\theta}^*$,i.e., $P(\boldsymbol{\theta}^*|\mathbf{y})$, using the output from a MH sampler. More specifically, let us denote the current state as $\boldsymbol{\theta}$. A possible candidate as future state $\mathbf{z} \sim \varphi(\mathbf{z}|\boldsymbol{\theta})$ (where $\varphi(\mathbf{z}|\boldsymbol{\theta})$ represents the proposal density used within MH), is accepted with probability $\alpha(\boldsymbol{\theta}, \mathbf{z}) = \min\left\{1, \frac{\pi(\mathbf{z}|\mathbf{y})\varphi(\boldsymbol{\theta}|\mathbf{z})}{\pi(\boldsymbol{\theta}|\mathbf{y})\varphi(\mathbf{z}|\boldsymbol{\theta})}\right\}$ [88, 61]. This is just an example of $\alpha(\boldsymbol{\theta}, \mathbf{z})$ that by construction the probability α satisfies the detailed balance condition [61, Section 2.4],[59], i.e.,

$$\alpha(\boldsymbol{\theta}, \mathbf{z})\varphi(\mathbf{z}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathbf{y}) = \alpha(\mathbf{z}, \boldsymbol{\theta})\varphi(\boldsymbol{\theta}|\mathbf{z})P(\mathbf{z}|\mathbf{y}). \tag{8.27}$$

By integrating in $\boldsymbol{\theta}$ both sides, we obtain

$$\begin{aligned}\int_\Theta \alpha(\boldsymbol{\theta}, \mathbf{z})\varphi(\mathbf{z}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} &= \int_\Theta \alpha(\mathbf{z}, \boldsymbol{\theta})\varphi(\boldsymbol{\theta}|\mathbf{z})P(\mathbf{z}|\mathbf{y})d\boldsymbol{\theta}, \\ &= P(\mathbf{z}|\mathbf{y}) \int_\Theta \alpha(\mathbf{z}, \boldsymbol{\theta})\varphi(\boldsymbol{\theta}|\mathbf{z})d\boldsymbol{\theta},\end{aligned}$$

hence finally we can solve with respect to $P(\mathbf{z}|\mathbf{y})$ obtaining

$$P(\mathbf{z}|\mathbf{y}) = \frac{\int_\Theta \alpha(\boldsymbol{\theta}, \mathbf{z})\varphi(\mathbf{z}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}}{\int_\Theta \alpha(\mathbf{z}, \boldsymbol{\theta})\varphi(\boldsymbol{\theta}|\mathbf{z})d\boldsymbol{\theta}}. \tag{8.28}$$

This suggests the following estimate of $P(\boldsymbol{\theta}^*|\mathbf{y})$ at a specific point $\boldsymbol{\theta}^*$ (note that $\boldsymbol{\theta}^*$ plays the role of $\mathbf{z}$ in the equation above),

$$\widehat{P}(\boldsymbol{\theta}^*|\mathbf{y}) = \frac{\frac{1}{N_1}\sum_{i=i}^{N_1} \alpha(\boldsymbol{\theta}_i, \boldsymbol{\theta}^*)\varphi(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i)}{\frac{1}{N_2}\sum_{j=1}^{N_2} \alpha(\boldsymbol{\theta}^*, \mathbf{v}_j)}, \qquad \{\boldsymbol{\theta}_i\}_{i=1}^{N_1} \sim P(\boldsymbol{\theta}|\mathbf{y}), \;\; \{\mathbf{v}_j\}_{j=1}^{N_2} \sim \varphi(\boldsymbol{\theta}|\boldsymbol{\theta}^*). \tag{8.29}$$

The same outputs of the MH scheme can be considered as $\{\boldsymbol{\theta}_i\}_{i=1}^{N_1}$. The final estimator is again $\widehat{Z} = \frac{\pi(\boldsymbol{\theta}^*|\mathbf{y})}{\widehat{P}(\boldsymbol{\theta}^*|\mathbf{y})}$, i.e.,

$$\widehat{Z} = \frac{\pi(\boldsymbol{\theta}^*|\mathbf{y})\frac{1}{N_2}\sum_{j=1}^{N_2}\alpha(\boldsymbol{\theta}^*, \mathbf{v}_j)}{\frac{1}{N_1}\sum_{i=i}^{N_1}\alpha(\boldsymbol{\theta}_i, \boldsymbol{\theta}^*)\varphi(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i)}, \qquad \{\boldsymbol{\theta}_i\}_{i=1}^{N_1} \sim P(\boldsymbol{\theta}|\mathbf{y}), \;\; \{\mathbf{v}_j\}_{j=1}^{N_2} \sim \varphi(\boldsymbol{\theta}|\boldsymbol{\theta}^*). \tag{8.30}$$

The point $\boldsymbol{\theta}^*$ is usually chosen in an high probability region. Interesting discussions are contained in [75], where the authors also show that this estimator is related to bridge sampling idea described in Section 8.4.2. For more details, see Section 8.4.2.

8.3.5. Interpolative approaches

Another possibility is to approximate Z by substituting the true $\pi(\boldsymbol{\theta}|\mathbf{y})$ with interpolation or a regression function $\widehat{\pi}(\boldsymbol{\theta}|\mathbf{y})$ in the integral (8.4). For simplicity, we focus on the interpolation case, but all the considerations can be easily extended for a regression scenario. Given a set of nodes $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\} \subset \Theta$ and N nonlinear functions $k(\boldsymbol{\theta}, \boldsymbol{\theta}') : \Theta \times \Theta \to \mathbb{R}$ chosen in advance by the user (generally, centered around $\boldsymbol{\theta}'$), we can build the interpolant of unnormalized posterior $\pi(\boldsymbol{\theta}|\mathbf{y})$ as follows

$$\widehat{\pi}(\boldsymbol{\theta}|\mathbf{y}) = \sum_{i=1}^{N}\beta_i k(\boldsymbol{\theta}, \boldsymbol{\theta}_i), \tag{8.31}$$

where $\beta_i \in \mathbb{R}$ and the subindex u denotes that is an approximation of the unnormalized function $\pi(\boldsymbol{\theta}|\mathbf{y})$. The coefficients β_i are chosen such that $\widehat{\pi}_u(\boldsymbol{\theta}|\mathbf{y})$ interpolates the points $\{\boldsymbol{\theta}_n, \pi(\boldsymbol{\theta}_n|\mathbf{y})\}$, that is, $\widehat{\pi}(\boldsymbol{\theta}_n|\mathbf{y}) = \pi(\boldsymbol{\theta}_n|\mathbf{y})$. Then, we desire that

$$\sum_{i=1}^{N}\beta_i k(\boldsymbol{\theta}_n, \boldsymbol{\theta}_i) = \pi(\boldsymbol{\theta}_n|\mathbf{y}),$$

for all $n = 1, ..., N$. Hence, we can write a $N \times N$ linear system where the β_i are the N unknowns, i.e.,

$$\begin{pmatrix} k(\boldsymbol{\theta}_1, \boldsymbol{\theta}_1) & k(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) & \ldots & k(\boldsymbol{\theta}_1, \boldsymbol{\theta}_N) \\ k(\boldsymbol{\theta}_2, \boldsymbol{\theta}_1) & k(\boldsymbol{\theta}_2, \boldsymbol{\theta}_2) & \ldots & k(\boldsymbol{\theta}_2, \boldsymbol{\theta}_N) \\ \vdots & & \ddots & \vdots \\ k(\boldsymbol{\theta}_N, \boldsymbol{\theta}_1) & k(\boldsymbol{\theta}_N, \boldsymbol{\theta}_2) & \ldots & k(\boldsymbol{\theta}_N, \boldsymbol{\theta}_N) \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{pmatrix} = \begin{pmatrix} \pi(\boldsymbol{\theta}_1|\mathbf{y}) \\ \pi(\boldsymbol{\theta}_2|\mathbf{y}) \\ \vdots \\ \pi(\boldsymbol{\theta}_N|\mathbf{y}) \end{pmatrix} \tag{8.32}$$

In matrix form, we have

$$\mathbf{K}\boldsymbol{\beta} = \mathbf{y}, \tag{8.33}$$

where $(\mathbf{K})_{i,j} = k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ and $\mathbf{y} = [\pi(\boldsymbol{\theta}_1|\mathbf{y}), \ldots, \pi(\boldsymbol{\theta}_N|\mathbf{y})]^\top$. Thus, the solution is $\boldsymbol{\beta} = \mathbf{K}^{-1}\mathbf{y}$. Now the interpolant $\widehat{\pi}_u(\boldsymbol{\theta}|\mathbf{y}) = \sum_{i=1}^{N}\beta_i k(\boldsymbol{\theta}, \boldsymbol{\theta}_i)$ can be used to approximate Z as follows

$$\widehat{Z} = \int_\Theta \widehat{\pi}_u(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} = \sum_{i=1}^{N}\beta_i \int_\Theta k(\boldsymbol{\theta}, \boldsymbol{\theta}_i)d\boldsymbol{\theta}. \tag{8.34}$$

If we are able to compute analytically $\int_\Theta k(\boldsymbol{\theta}, \boldsymbol{\theta}_i) d\boldsymbol{\theta}$, we have an approximation $\widehat{Z}$. Some suitable choices of $k(\cdot, \cdot)$ are rectangular, triangular and Gaussian functions. More specifically, if all the nonlinearities $k(\boldsymbol{\theta}, \boldsymbol{\theta}_i)$ are normalized (i.e. $\int_\Theta k(\boldsymbol{\theta}, \boldsymbol{\theta}_i) d\boldsymbol{\theta} = 1$), the approximation of Z is $\widehat{Z} = \sum_{i=1}^N \beta_i$. This approach is related to the so-called Bayesian quadrature (using Gaussian process approximation) [87] and the sticky proposal constructions within MCMC or rejection sampling algorithms [40, 38, 60, 73]. Adaptive schemes adding sequentially more nodes could be also considered, improving the approximation $\widehat{Z}$ [38, 60]. The quality of the interpolating approximation deteriorates as the dimension of $\boldsymbol{\theta}$ grows (see e.g. [6] for explicit error bounds).

8.4. Techniques based on IS

Most of the techniques for approximating the marginal likelihood are based on the importance sampling (IS) approach. Other methods are directly or indirectly related to the IS framework. In this sense, this section is the core of this survey. The standard IS scheme relies on the following equality,

$$Z = \int_\Theta \pi(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} = \mathbb{E}_{\bar{q}}\left[\frac{\pi(\boldsymbol{\theta}|\mathbf{y})}{\bar{q}(\boldsymbol{\theta})}\right] = \int_\Theta \frac{\pi(\boldsymbol{\theta}|\mathbf{y})}{\bar{q}(\boldsymbol{\theta})} \bar{q}(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{8.35}$$

$$= \int_\Theta \frac{\ell(\mathbf{y}|\boldsymbol{\theta}) g(\boldsymbol{\theta})}{\bar{q}(\boldsymbol{\theta})} \bar{q}(\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{8.36}$$

where $\bar{q}(\boldsymbol{\theta})$ is a simpler normalized proposal density, $\int_\Theta \bar{q}(\boldsymbol{\theta}) d\boldsymbol{\theta} = 1$.

IS version 1. Drawing N independent samples from proposal $\bar{q}(\boldsymbol{\theta})$, the *unbiased* IS estimator (denoted as IS vers-1) of Z is

$$\widehat{Z}_{IS\,1} = \frac{1}{N} \sum_{i=1}^N \frac{\pi(\boldsymbol{\theta}_i|\mathbf{y})}{\bar{q}(\boldsymbol{\theta}_i)} \tag{8.37}$$

$$= \frac{1}{N} \sum_{i=1}^N w_i, \tag{8.38}$$

$$= \frac{1}{N} \sum_{i=1}^N \frac{\ell(\mathbf{y}|\boldsymbol{\theta}_i) g(\boldsymbol{\theta}_i)}{\bar{q}(\boldsymbol{\theta}_i)} = \frac{1}{N} \sum_{i=1}^N \rho_i \ell(\mathbf{y}|\boldsymbol{\theta}_i), \qquad \{\boldsymbol{\theta}_i\}_{i=1}^N \sim \bar{q}(\boldsymbol{\theta}), \tag{8.39}$$

where $w_i = \frac{\pi(\boldsymbol{\theta}_i|\mathbf{y})}{\bar{q}(\boldsymbol{\theta}_i)}$ are the standard IS weights and $\rho_i = \frac{g(\boldsymbol{\theta}_i)}{\bar{q}(\boldsymbol{\theta}_i)}$.

Optimal proposal in IS vers-1. The optimal proposal, in terms of mean square error (MSE), in the standard IS scheme above is $\bar{q}^{\text{opt}}(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y})$.

IS version 2. An alternative IS estimator (denoted as IS vers-2) is given by, considering

a possibly unnormalized proposal pdf $q(\boldsymbol{\theta}) \propto \bar{q}(\boldsymbol{\theta})$ (the case $q(\boldsymbol{\theta}) = \bar{q}(\boldsymbol{\theta})$ is also included),

$$\widehat{Z}_{IS2} = \frac{1}{\sum_{n=1}^{N} \frac{g(\boldsymbol{\theta}_n)}{q(\boldsymbol{\theta}_n)}} \sum_{i=1}^{N} \frac{g(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)} \ell(\mathbf{y}|\boldsymbol{\theta}_i), \tag{8.40}$$

$$= \frac{1}{\sum_{n=1}^{N} \rho_n} \sum_{i=1}^{N} \rho_i \ell(\mathbf{y}|\boldsymbol{\theta}_i), \tag{8.41}$$

$$= \sum_{i=1}^{N} \bar{\rho}_i \ell(\mathbf{y}|\boldsymbol{\theta}_i), \qquad \{\boldsymbol{\theta}_i\}_{i=1}^{N} \sim \bar{q}(\boldsymbol{\theta}). \tag{8.42}$$

The estimator above is biased. However, it is a convex combination of likelihood values $\ell(\mathbf{y}|\boldsymbol{\theta}_i)$ since $\sum_{i=1}^{N} \bar{\rho}_i = 1$. Hence, in this case $\min_i \ell(\mathbf{y}|\boldsymbol{\theta}_i) \leq \widehat{Z} \leq \max_i \ell(\mathbf{y}|\boldsymbol{\theta}_i)$, i.e., the estimator fulfills the bounds of Z, shown Section 8.2.2. Moreover, the estimator allows the use of an unnormalized proposal pdf $q(\boldsymbol{\theta}) \propto \bar{q}(\boldsymbol{\theta})$ and $\rho_i = \frac{g(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)}$. For instance, one could consider $\bar{q}(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y})$, i.e., generate samples $\{\boldsymbol{\theta}_i\}_{i=1}^{N} \sim P(\boldsymbol{\theta}|\mathbf{y})$ by an MCMC algorithm and then evaluate $\rho_i = \frac{g(\boldsymbol{\theta}_i)}{\pi(\boldsymbol{\theta}_i|\mathbf{y})}$.

Optimal proposal in IS vers-2. The optimal proposal, in terms of MSE, for the IS vers-2 is $\bar{q}^{\mathbf{opt}}(\boldsymbol{\theta}) \propto |P(\boldsymbol{\theta}|\mathbf{y}) - g(\boldsymbol{\theta})|$.

Table 8.3 summarizes the IS estimators and shows some important special cases that will be described in the next section.

Table 8.3: IS estimators Eqs. (8.37)-(8.40) and relevant special cases.

$\widehat{Z}_{IS1} = \frac{1}{N}\sum_{i=1}^{N} \frac{g(\boldsymbol{\theta}_i)}{\bar{q}(\boldsymbol{\theta}_i)} \ell(\mathbf{y}\|\boldsymbol{\theta}_i) = \frac{1}{N}\sum_{i=1}^{N} \rho_i \ell(\mathbf{y}\|\boldsymbol{\theta}_i), \quad \rho_i = \frac{g(\boldsymbol{\theta}_i)}{\bar{q}(\boldsymbol{\theta}_i)}$					
Name	Estimator	$q(\boldsymbol{\theta})$	$\bar{q}(\boldsymbol{\theta})$	Need of MCMC	Unbiased
Naive Monte Carlo	$\frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{y}\|\boldsymbol{\theta}_i)$	$g(\boldsymbol{\theta})$	$g(\boldsymbol{\theta})$	—	✓
$\widehat{Z}_{IS2} = \frac{1}{\sum_{n=1}^{N} \frac{g(\boldsymbol{\theta}_n)}{q(\boldsymbol{\theta}_n)}} \sum_{i=1}^{N} \frac{g(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)} \ell(\mathbf{y}\|\boldsymbol{\theta}_i) = \sum_{i=1}^{N} \bar{\rho}_i \ell(\mathbf{y}\|\boldsymbol{\theta}_i)$					
Name	Estimator	$q(\boldsymbol{\theta})$	$\bar{q}(\boldsymbol{\theta})$	Need of MCMC	Unbiased
Naive Monte Carlo	$\frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{y}\|\boldsymbol{\theta}_i)$	$g(\boldsymbol{\theta})$	$g(\boldsymbol{\theta})$	—	✓
Harmonic mean	$\left(\frac{1}{N}\sum_{i=1}^{N} \frac{1}{\ell(\mathbf{y}\|\boldsymbol{\theta}_i)}\right)^{-1}$	$\pi(\boldsymbol{\theta}\|\mathbf{y})$	$P(\boldsymbol{\theta}\|\mathbf{y})$	✓	—

Different sub-families of IS schemes are commonly used for computing normalizing constants [14, chapter 5]. A first approach uses draws from a proposal density $\bar{q}(\boldsymbol{\theta})$ that is completely known (i.e. direct sampling and evaluate). Sophisticated choices of $\bar{q}(\boldsymbol{\theta})$ frequently imply the use of MCMC algorithms to sample from $\bar{q}(\boldsymbol{\theta})$ and that we can only evaluate $q(\boldsymbol{\theta}) \propto \bar{q}(\boldsymbol{\theta})$. The one-proposal approach is described in Section 8.4.1. A second class is formed by methods which use more than one proposal density or a mixture

of them (see Sections 8.4.2, 8.4.3 and 8.5). Moreover, **adaptive importance sampling (AIS)** schemes are often designed, where the proposal (or the cloud of proposals) is improved during some iterations, in some way such that $\bar{q}_t(\boldsymbol{\theta})$ (where t is an iteration index) becomes closer and closer to the optima proposal $q^{\text{opt}}(\boldsymbol{\theta})$. For more details, see the reviews in [7]. Some AIS methods, obtained combining MCMC and IS approaches, are described in Section 8.5.

8.4.1. Techniques using draws from one proposal density

In this section, all the techniques are IS schemes which use a unique proposal pdf, and are based on the identity Eq. (8.35). The techniques differ in the choice of $\bar{q}(\boldsymbol{\theta})$. Recall that the optimal proposal choice for IS vers-1 is $\bar{q}(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y}) = \frac{1}{Z}\pi(\boldsymbol{\theta}|\mathbf{y})$. This choice is clearly difficult for two reasons: (a) we have to draw from P and (b) we do not know Z, hence we cannot evaluate $\bar{q}(\boldsymbol{\theta})$ but only $q(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\mathbf{y})$ (where $q(\boldsymbol{\theta}) \propto \bar{q}(\boldsymbol{\theta})$). However, there are some methods based on this idea, as shown in the following. The techniques below are enumerated in an increasing order of complexity.

Naive Monte Carlo (arithmetic mean estimator). It is straightforward to note that the integral above can be expressed as $Z = \mathbb{E}_g[\ell(\mathbf{y}|\boldsymbol{\theta})]$, then we can draw N samples $\{\boldsymbol{\theta}_i\}_{i=1}^N$ from the prior $g(\boldsymbol{\theta})$ and compute the following estimator

$$\widehat{Z} = \frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{y}|\boldsymbol{\theta}_i), \qquad \{\boldsymbol{\theta}_i\}_{i=1}^N \sim g(\boldsymbol{\theta}). \tag{8.43}$$

Namely a simple average of the likelihoods of a sample from the prior. Note that $\widehat{Z}$ will be very inefficient (large variance) if the posterior is much more concentrated than the prior (i.e., small overlap between likelihood and prior pdfs). Therefore, alternatives have been proposed, see below. It is a special case of the IS estimator with the choice $\bar{q}(\boldsymbol{\theta}) = g(\boldsymbol{\theta})$ (i.e., the proposal pdf is the prior).

Harmonic mean (HM) estimators. The HM estimator can be directly derived from the following expected value,

$$\mathbb{E}_P\left[\frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta})}\right] = \int_\Theta \frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta})} P(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta}, \tag{8.44}$$

$$= \frac{1}{Z}\int_\Theta \frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta})} \ell(\mathbf{y}|\boldsymbol{\theta}) g(\boldsymbol{\theta}) d\boldsymbol{\theta} = \frac{1}{Z}\int_\Theta g(\boldsymbol{\theta}) d\boldsymbol{\theta} = \frac{1}{Z}. \tag{8.45}$$

The main idea is again to use the posterior itself as proposal. Since direct sampling from $P(\boldsymbol{\theta}|\mathbf{y})$ is generally impossible, this task requires the use of MCMC algorithms. Thus, the HM estimator is

$$\widehat{Z} = \left(\frac{1}{N}\sum_{i=1}^{N} \frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta}_i)}\right)^{-1} = \frac{1}{\frac{1}{N}\sum_{i=1}^{N} \frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta}_i)}}, \quad \{\boldsymbol{\theta}_i\}_{i=1}^N \sim P(\boldsymbol{\theta}|\mathbf{y}) \text{ (via MCMC)}. \tag{8.46}$$

The HM estimator converges almost surely to the correct value, but the variance of $\widehat{Z}$ is often high and possibly infinite. [14] The HM estimator is a special case of Reverse Importance Sampling (RIS) below.

Reverse Importance Sampling (RIS). The RIS scheme [33], also known as *reciprocal* IS, can be derived from the identity

$$\frac{1}{Z} = \mathbb{E}_P\left[\frac{f(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}|\mathbf{y})}\right] = \int_\Theta \frac{f(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}|\mathbf{y})} P(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \tag{8.47}$$

where we consider an auxiliary normalized function $f(\boldsymbol{\theta})$, i.e., $\int_\Theta f(\boldsymbol{\theta})d\boldsymbol{\theta} = 1$. Then, one could consider the estimator

$$\widehat{Z} = \left(\frac{1}{N}\sum_{i=1}^N \frac{f(\boldsymbol{\theta}_i)}{\pi(\boldsymbol{\theta}_i|\mathbf{y})}\right)^{-1} = \left(\frac{1}{N}\sum_{i=1}^N \frac{f(\boldsymbol{\theta}_i)}{\ell(\mathbf{y}|\boldsymbol{\theta}_i)g(\boldsymbol{\theta}_i)}\right)^{-1}, \quad \boldsymbol{\theta}_i \sim P(\boldsymbol{\theta}|\mathbf{y}) \text{ (via MCMC)} \tag{8.48}$$

The estimator above is consistent but biased. Indeed, the expression $\frac{1}{N}\sum_{i=1}^N \frac{f(\boldsymbol{\theta}_i)}{\pi(\boldsymbol{\theta}_i|\mathbf{y})}$ is a unbiased estimator of $1/Z$, but $\widehat{Z}$ in the Eq. (8.48) is not an unbiased estimator of Z. Note that $P(\boldsymbol{\theta}|\mathbf{y})$ plays the role of importance density from which we need to draw from. Therefore, another sampling technique must be used (such as a MCMC method) in order to generate samples from $P(\boldsymbol{\theta}|\mathbf{y})$. In this case, we do not need samples from $f(\boldsymbol{\theta})$, although its choice affects the precision of the approximation. Unlike in the standard IS approach, $f(\boldsymbol{\theta})$ must have lighter tails than $\pi(\boldsymbol{\theta}|\mathbf{y}) = \ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})$. For further details, see the example in Section 8.8.1. Finally, note that the HM estimator is a special case of RIS when $f(\boldsymbol{\theta}) = g(\boldsymbol{\theta})$ in Eq. (8.48). In [89], the authors propose taking $f(\boldsymbol{\theta})$ that is uniform in a high posterior density region whereas, in [104], they consider taking $f(\boldsymbol{\theta})$ to be a piecewise constant function.

The pre-umbrella estimators

All the estimators that we have seen so far can be unified within a common formulation, considering the more general problem of estimating a ratio of two normalizing constants c_1/c_2, where $c_i = \int q_i(\boldsymbol{\theta})d\boldsymbol{\theta}$ and $\bar{q}_i(\boldsymbol{\theta}) = q_i(\boldsymbol{\theta})/c_i$, $i = 1, 2$. Assuming we can evaluate both $q_1(\boldsymbol{\theta})$, $q_2(\boldsymbol{\theta})$, and draw samples from one of them, say $\bar{q}_2(\boldsymbol{\theta})$, the importance sampling estimator of ratio c_1/c_2 is

$$\frac{c_1}{c_2} = \mathbb{E}_{\bar{q}_2}\left[\frac{q_1(\boldsymbol{\theta})}{q_2(\boldsymbol{\theta})}\right] \approx \frac{1}{N}\sum_{i=1}^N \frac{q_1(\boldsymbol{\theta}_i)}{q_2(\boldsymbol{\theta}_i)}, \quad \{\boldsymbol{\theta}_i\}_{i=1}^N \sim \bar{q}_2(\boldsymbol{\theta}). \tag{8.49}$$

Remark 6. *The relative MSE (rel-MSE) of* (8.49)*, in estimation of the ratio* $r = \frac{c_1}{c_2}$*, i.e., rel-MSE* $= \frac{\mathbb{E}[(\widehat{r}-r)^2]}{r^2}$*, is given by rel-MSE* $= \frac{1}{N}\chi^2(\bar{q}_1||\bar{q}_2)$*, where* $\chi^2(\bar{q}_1||\bar{q}_2)$ *is the Pearson*

[14] See the comments of Radford Neal's blog, https://radfordneal.wordpress.com/2008/08/17/the-harmonic-mean-of-the-likelihood-worst-monte-carlo-method-ever/, where R. Neal defines the HM estimator as "the worst estimator ever".

divergence between $\bar{q}_1$ and $\bar{q}_2$ [13].

This framework includes almost all the estimators discussed so far in this section, as shown in Table 8.4. However, the IS vers-2 estimator is not a special case of Eq. (8.49).

Table 8.4: Summary of techniques considering the expression (8.49).

Name	$q_1(\boldsymbol{\theta})$	$q_2(\boldsymbol{\theta})$	c_1	c_2	Proposal pdf $\bar{q}_2(\boldsymbol{\theta})$	c_1/c_2
IS vers-1	$\pi(\boldsymbol{\theta}\|\mathbf{y})$	$\bar{q}(\boldsymbol{\theta})$	Z	1	$\bar{q}(\boldsymbol{\theta})$	Z
Naive Monte Carlo	$\pi(\boldsymbol{\theta}\|\mathbf{y})$	$g(\boldsymbol{\theta})$	Z	1	$g(\boldsymbol{\theta})$	Z
Harmonic mean	$g(\boldsymbol{\theta})$	$\pi(\boldsymbol{\theta}\|\mathbf{y})$	1	Z	$P(\boldsymbol{\theta}\|\mathbf{y})$	$1/Z$
RIS	$f(\boldsymbol{\theta})$	$\pi(\boldsymbol{\theta}\|\mathbf{y})$	1	Z	$P(\boldsymbol{\theta}\|\mathbf{y})$	$1/Z$

Below we consider an extension of Eq. (8.49) where an additional density $\bar{q}_3(\boldsymbol{\theta})$ is employed for generating samples.

Umbrella Sampling (a.k.a. ratio importance sampling)

The IS estimator of c_1/c_2 given in Eq. (8.49) may be inefficient when there is little overlap between $\bar{q}_1(\boldsymbol{\theta})$ and $\bar{q}_2(\boldsymbol{\theta})$, i.e., when $\int_\Theta \bar{q}_1(\boldsymbol{\theta})\bar{q}_2(\boldsymbol{\theta})d\boldsymbol{\theta}$ is small. Umbrella sampling (originally proposed in the computational physics literature, [100]; also studied under the name ratio importance sampling in [13]) is based on the identity

$$\frac{c_1}{c_2} = \frac{c_1/c_3}{c_2/c_3} = \frac{\mathbb{E}_{\bar{q}_3}\left[\frac{q_1(\boldsymbol{\theta})}{q_3(\boldsymbol{\theta})}\right]}{\mathbb{E}_{\bar{q}_3}\left[\frac{q_2(\boldsymbol{\theta})}{q_3(\boldsymbol{\theta})}\right]} \approx \frac{\sum_{i=1}^N \frac{q_1(\boldsymbol{\theta}_i)}{q_3(\boldsymbol{\theta}_i)}}{\sum_{i=1}^N \frac{q_2(\boldsymbol{\theta}_i)}{q_3(\boldsymbol{\theta}_i)}}, \quad \{\boldsymbol{\theta}_i\}_{i=1}^N \sim \bar{q}_3(\boldsymbol{\theta}) \tag{8.50}$$

where $\bar{q}_3(\boldsymbol{\theta}) \propto q_3(\boldsymbol{\theta})$ represents a *middle* density. A good choice of $\bar{q}_3(\boldsymbol{\theta})$ should have large overlaps with both $\bar{q}_i(\boldsymbol{\theta})$, $i = 1, 2$. The performance of umbrella sampling clearly depends on the choice of $\bar{q}_3(\boldsymbol{\theta})$. Note that, when $\bar{q}_3 = \bar{q}_2$, we recover Eq. (8.49).

Optimal umbrella proposal. The optimal umbrella sampling density $\bar{q}_3^{\text{opt}}(\boldsymbol{\theta})$, that minimizes the asymptotic relative mean-square error, is

$$\bar{q}_3^{\text{opt}}(\boldsymbol{\theta}) = \frac{|\bar{q}_1(\boldsymbol{\theta}) - \bar{q}_2(\boldsymbol{\theta})|}{\int |\bar{q}_1(\boldsymbol{\theta}') - \bar{q}_2(\boldsymbol{\theta}')|d\boldsymbol{\theta}'} = \frac{|q_1(\boldsymbol{\theta}) - \frac{c_1}{c_2}q_2(\boldsymbol{\theta})|}{\int |q_1(\boldsymbol{\theta}') - \frac{c_1}{c_2}q_2(\boldsymbol{\theta}')|d\boldsymbol{\theta}'}. \tag{8.51}$$

Remark 7. *The rel-MSE in estimation of the ratio $\frac{c_1}{c_2}$ of the optimal umbrella estimator, with N great enough, is given by rel-MSE $\approx \frac{1}{N}L_1^2(\bar{q}_1, \bar{q}_2)$, where $L_1^2(\bar{q}_1, \bar{q}_2)$ denotes the L_1-distance between $\bar{q}_1$ and $\bar{q}_2$ [13, Theorem 3.2]. Moreover, since $L_1^2(\bar{q}_1, \bar{q}_2) \leq \chi^2(\bar{q}_1\|\bar{q}_2)$, the optimal umbrella estimator is asymptotically more efficient than the estimator* (8.49) *[12, Sect. 3].*

Two-stage umbrella sampling. Since this $\bar{q}_3^{\text{opt}}(\boldsymbol{\theta})$ depends on the unknown ratio $\frac{c_1}{c_2}$, it

is not available for a direct use. The following two-stage procedure is often used in practice:

1. *Stage 1*: Draw N_1 samples from an arbitrary density $\bar{q}_3^{(1)}(\boldsymbol{\theta})$ and use them to obtain

$$\widehat{r}^{(1)} = \frac{\sum_{i=1}^{N_1} \frac{q_1(\boldsymbol{\theta}_i)}{q_3^{(1)}(\boldsymbol{\theta}_i)}}{\sum_{i=1}^{N_1} \frac{q_2(\boldsymbol{\theta}_i)}{q_3^{(1)}(\boldsymbol{\theta}_i)}}, \quad \{\boldsymbol{\theta}_i\}_{i=1}^{N_1} \sim \bar{q}_3^{(1)}(\boldsymbol{\theta}). \tag{8.52}$$

   and define

$$\bar{q}_3^{(2)}(\boldsymbol{\theta}) \propto |q_1(\boldsymbol{\theta}) - \widehat{r}^{(1)} q_2(\boldsymbol{\theta})|. \tag{8.53}$$

2. *Stage 2*: Draw N_2 samples from $\bar{q}_3^{(2)}(\boldsymbol{\theta})$ via MCMC and define the umbrella sampling estimator $\widehat{r}^{(2)}$ of $\frac{c_1}{c_2}$ as follows

$$\widehat{r}^{(2)} = \frac{\sum_{i=1}^{n_2} \frac{q_1(\boldsymbol{\theta}_i)}{q_3^{(2)}(\boldsymbol{\theta}_i)}}{\sum_{i=1}^{n_2} \frac{q_2(\boldsymbol{\theta}_i)}{q_3^{(2)}(\boldsymbol{\theta}_i)}}, \quad \{\boldsymbol{\theta}_i\}_{i=1}^{n_2} \sim \bar{q}_3^{(2)}(\boldsymbol{\theta}). \tag{8.54}$$

Remark 8. *The number of stages could be increased considering, at each t-th stage, the proposal* $\bar{q}_3^{(t)}(\boldsymbol{\theta}) \propto |q_1(\boldsymbol{\theta}) - \widehat{r}^{(t-1)} q_2(\boldsymbol{\theta})|$ *and obtaining a new estimation* $\widehat{r}^{(t)}$*. In this case, we have an umbrella scheme with adaptive proposal* $\bar{q}_3^{(t)}(\boldsymbol{\theta})$*.*

Umbrella for Z: the self-normalized Importance Sampling (Self-IS)

Here, we describe an important special case of the umbrella sampling approach. Considering the umbrella identity (8.50) an setting $q_1(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\mathbf{y})$, $q_2(\boldsymbol{\theta}) = \bar{q}_2(\boldsymbol{\theta}) = f(\boldsymbol{\theta})$, $c_1 = Z$, $c_2 = 1$ and $c_3 \in \mathbb{R}$, we obtain

$$\widehat{Z} = \frac{1}{\sum_{i=1}^{N} \frac{f(\boldsymbol{\theta}_i)}{q_3(\boldsymbol{\theta}_i)}} \sum_{i=1}^{N} \frac{\pi(\boldsymbol{\theta}_i|\mathbf{y})}{q_3(\boldsymbol{\theta}_i)}. \qquad \{\boldsymbol{\theta}_i\}_{i=1}^{N} \sim \bar{q}_3(\boldsymbol{\theta}). \tag{8.55}$$

which is called the *self-normalized IS* (Self-IS) estimator. Note that $f(\boldsymbol{\theta})$ is an auxiliary normalized pdf, but we draw samples from $\bar{q}_3(\boldsymbol{\theta})$. In order to understand the reason of its name is interesting to derive it with standard IS arguments. Let us consider that our proposal $q(\boldsymbol{\theta})$ in the standard IS scheme is not normalized, and we can evaluate it up to a normalizing constant $q(\boldsymbol{\theta}) \propto \bar{q}(\boldsymbol{\theta})$. We also denote $c = \int_\Theta q(\boldsymbol{\theta}) d\boldsymbol{\theta}$. Note that this also occurs in the ideal case of using $\bar{q}(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y}) = \frac{1}{Z}\pi(\boldsymbol{\theta}|\mathbf{y})$ where $c = Z$ and $q(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\mathbf{y})$. In this case, we have

$$\frac{\widehat{Z}}{c} = \frac{1}{N} \sum_{i=1}^{N} \frac{\pi(\boldsymbol{\theta}_i|\mathbf{y})}{q(\boldsymbol{\theta}_i)}, \qquad \{\boldsymbol{\theta}_i\}_{i=1}^{N} \sim \bar{q}(\boldsymbol{\theta}). \tag{8.56}$$

Therefore, we need an additional estimation of c. We can also use IS for this goal, considering a new normalized reference function $f(\boldsymbol{\theta})$, i.e., $\int_\Theta f(\boldsymbol{\theta}) d\boldsymbol{\theta} = 1$. Now,

$$\frac{1}{c} = E_{\bar{q}}\left[\frac{f(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}\right] = \int_\Theta \frac{f(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \bar{q}(\boldsymbol{\theta}) d\boldsymbol{\theta} \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)}, \qquad \{\boldsymbol{\theta}_i\}_{i=1}^{N} \sim \bar{q}(\boldsymbol{\theta}). \tag{8.57}$$

Replacing (8.57) into (8.56), we obtain the self-normalized IS estimator in Eq. (8.55), i.e., $\widehat{Z} = \frac{1}{\sum_{i=1}^{N} \frac{f(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)}} \sum_{i=1}^{N} \frac{\pi(\boldsymbol{\theta}_i|\mathbf{y})}{q(\boldsymbol{\theta}_i)}$ with $\{\boldsymbol{\theta}_i\}_{i=1}^{N} \sim \bar{q}(\boldsymbol{\theta})$.
The HM estimator is also a special case of Self-IS setting again $f(\boldsymbol{\theta}) = g(\boldsymbol{\theta})$ and $\bar{q}(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y})$, so that $q(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\mathbf{y})$. Moreover, the RIS estimator is a special case of the Self-IS estimator above when $\bar{q}(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y})$ and $q(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\mathbf{y})$.

Optimal self-IS (O-Self-IS). Since the Self-IS estimator is a special case of umbrella sampling, the optimal proposal in this case is $\bar{q}^{\text{opt}}(\boldsymbol{\theta}) \propto |P(\boldsymbol{\theta}|\mathbf{y}) - f(\boldsymbol{\theta})|$, and the optimal estimator is

$$\widehat{Z}_{\text{O-Self-IS}} = \frac{\sum_{i=1}^{N} \frac{\pi(\boldsymbol{\theta}_i)}{|P(\boldsymbol{\theta}_i|\mathbf{y}) - f(\boldsymbol{\theta}_i)|}}{\sum_{i=1}^{N} \frac{f(\boldsymbol{\theta}_i)}{|P(\boldsymbol{\theta}_i|\mathbf{y}) - f(\theta_i)|}}, \quad \boldsymbol{\theta}_i \sim \bar{q}^{\text{opt}}(\boldsymbol{\theta}) \propto |P(\boldsymbol{\theta}|\mathbf{y}) - f(\boldsymbol{\theta})|. \tag{8.58}$$

Since the density cannot be evaluated (and also is not easy to draw from), this estimator is not of direct use and we need to resort to the two-stage procedure that we discussed above. Due to Remark 7, the O-Self-IS estimator is asymptotically more efficient than IS vers-1 estimator using $f(\boldsymbol{\theta})$ as proposal, i.e., drawing samples from $\bar{q}(\boldsymbol{\theta}) = f(\boldsymbol{\theta})$.

Summary

The more general expressions are the two identities (8.49)-(8.50) for estimating a ratio of normalizing constants $\frac{c_1}{c_2}$. The umbrella identity (8.50) is the more general since three densities are involved, and contains the Eq. (8.49) as special case when $q_3(\boldsymbol{\theta}) = q_2(\boldsymbol{\theta})$. The Self-IS estimator coincides with the umbrella estimator when we approximate only one normalizing constant, Z (i.e., for $\frac{c_1}{c_2} = Z$). Therefore, regarding the estimation of only one constant Z, the Self-IS estimator has the more general form and includes the rest of estimators as special cases. All these connections are summarized in Table 8.5. Finally, Table 8.6 provides another summary of the one-proposal estimators of Z. Note that in the standard IS estimator the option $\bar{q}(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y})$ is not feasible, whereas it is possible for its second version.

In the next section, we discuss a generalization of Eq. (8.49) for the case where we use samples from both $\bar{q}_1(\boldsymbol{\theta})$ and $\bar{q}_2(\boldsymbol{\theta})$.

8.4.2. Techniques using draws from two proposal densities

In the previous section, we considered estimators of Z that use samples drawn from a single proposal density. More specifically, we have described several IS schemes using a generic pdf $\bar{q}(\boldsymbol{\theta})$ *or* $P(\boldsymbol{\theta}|\mathbf{y})$ as proposal density. In this section, we introduce schemes where $\bar{q}(\boldsymbol{\theta})$ *and* $P(\boldsymbol{\theta}|\mathbf{y})$ are employed jointly. More generally, we consider estimators of a ratio of constants, $\frac{c_2}{c_1}$, that employ samples from two proposal densities, denoted as $\bar{q}_i(\boldsymbol{\theta}) = \frac{q_i(\boldsymbol{\theta})}{c_i}, i = 1, 2$. Note that drawing N_1 samples from $\bar{q}_1(\boldsymbol{\theta})$ and N_2 samples from $\bar{q}_2(\boldsymbol{\theta})$ is equivalent to sampling by a *deterministic mixture* approach from the mixture

Table 8.5: Summary of techniques considering the umbrella sampling identity (8.50) for computing $\frac{c_1}{c_2} = Z$. Note that Self-IS has the more general form and includes the rest of estimators as special cases.

For estimating a generic ratio $c1/c2$

Umbrella	$q_1(\boldsymbol{\theta})$	$q_2(\boldsymbol{\theta})$	$q_3(\boldsymbol{\theta})$	c_1	c_2	c_3	sampling from $\bar{q}_3(\boldsymbol{\theta})$
Eq. (8.49) - ($q_3 = q_2$)	$q_1(\boldsymbol{\theta})$	$q_2(\boldsymbol{\theta})$	$q_2(\boldsymbol{\theta})$	c_1	c_2	c_2	sampling from $\bar{q}_2(\boldsymbol{\theta})$

For estimating Z

Self-IS	$\pi(\boldsymbol{\theta}\|\mathbf{y})$	$f(\boldsymbol{\theta})$	$q(\boldsymbol{\theta})$	Z	1	c_3	$\bar{q}(\boldsymbol{\theta})$

Special cases of Self-IS

Naive Monte Carlo	$\pi(\boldsymbol{\theta}\|\mathbf{y})$	$g(\boldsymbol{\theta})$	$g(\boldsymbol{\theta})$			1	$g(\boldsymbol{\theta})$
Harmonic Mean	$\pi(\boldsymbol{\theta}\|\mathbf{y})$	$g(\boldsymbol{\theta})$	$\pi(\boldsymbol{\theta}\|\mathbf{y})$			Z	$P(\boldsymbol{\theta}\|\mathbf{y})$
RIS	$\pi(\boldsymbol{\theta}\|\mathbf{y})$	$f(\boldsymbol{\theta})$	$\pi(\boldsymbol{\theta}\|\mathbf{y})$	Z	1	Z	$P(\boldsymbol{\theta}\|\mathbf{y})$
IS vers-1; Eq. (8.37)	$\pi(\boldsymbol{\theta}\|\mathbf{y})$	$\bar{q}(\boldsymbol{\theta})$	$\bar{q}(\boldsymbol{\theta})$			1	$\bar{q}(\boldsymbol{\theta})$
IS vers-2; Eq. (8.40)	$\pi(\boldsymbol{\theta}\|\mathbf{y})$	$g(\boldsymbol{\theta})$	$\bar{q}(\boldsymbol{\theta})$			1	$\bar{q}(\boldsymbol{\theta})$

Table 8.6: One-proposal estimators of Z

Name	Estimator	Proposal pdf	Need of MCMC	Unbiased
IS vers-1	$\frac{1}{N}\sum_{i=1}^{N}\rho_i\ell(\mathbf{y}\|\boldsymbol{\theta}_i)$	Generic, $\bar{q}(\boldsymbol{\theta})$	—	✓
IS vers-2	$\sum_{i=1}^{N}\bar{\rho}_i\ell(\mathbf{y}\|\boldsymbol{\theta}_i)$	Generic, $\bar{q}(\boldsymbol{\theta})$	no, if $\bar{q}(\boldsymbol{\theta}) \neq P(\boldsymbol{\theta}\|\mathbf{y})$	—
Naive MC	$\frac{1}{N}\sum_{i=1}^{N}\ell(\mathbf{y}\|\boldsymbol{\theta}_i)$	Prior, $g(\boldsymbol{\theta})$	—	✓
Harmonic mean	$\left(\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\ell(\mathbf{y}\|\boldsymbol{\theta}_i)}\right)^{-1}$	Posterior, $P(\boldsymbol{\theta}\|\mathbf{y})$	✓	—
RIS	$\left(\frac{1}{N}\sum_{i=1}^{N}\frac{f(\boldsymbol{\theta}_i)}{\pi(\boldsymbol{\theta}_i\|\mathbf{y})}\right)^{-1}$	Posterior, $P(\boldsymbol{\theta}\|\mathbf{y})$	✓	—
Self-IS	$\left(\sum_{i=1}^{N}\frac{f(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)}\right)^{-1}\sum_{i=1}^{N}\frac{\pi(\boldsymbol{\theta}_i\|\mathbf{y})}{q(\boldsymbol{\theta}_i)}$	Generic, $\bar{q}(\boldsymbol{\theta})$	no, if $\bar{q}(\boldsymbol{\theta}) \neq P(\boldsymbol{\theta}\|\mathbf{y})$	—

$\bar{q}_{mix}(\boldsymbol{\theta}) = \frac{N_1}{N_1+N_2}\bar{q}_1(\boldsymbol{\theta}) + \frac{N_2}{N_1+N_2}\bar{q}_2(\boldsymbol{\theta})$, i.e., a single density defined as mixture of two pdfs [28]. Thus, methods drawing from a mixture of two pdfs as $\bar{q}_{mix}(\boldsymbol{\theta})$, are also considered in this section.

Bridge sampling identity

All the techniques, that we will describe below, are based on the following *bridge sampling* identity [74],

$$\frac{c_1}{c_2} = \frac{\mathbb{E}_{\bar{q}_2}[q_1(\boldsymbol{\theta})\alpha(\boldsymbol{\theta})]}{\mathbb{E}_{\bar{q}_1}[q_2(\boldsymbol{\theta})\alpha(\boldsymbol{\theta})]}. \tag{8.59}$$

where $\alpha(\boldsymbol{\theta})$ is an arbitrary function defined on the intersection of the supports of $\bar{q}_1$ and $\bar{q}_2$. Note that the expression above is an extension of the Eq. (8.49). Indeed, taking $\alpha(\boldsymbol{\theta}) = \frac{1}{q_2(\boldsymbol{\theta})}$, we recover Eq. (8.49). The identity in Eq. (8.59) and the umbrella identity

in Eq. (8.50) are both useful when $\bar{q}_1$ and $\bar{q}_2$ have little overlap, i.e., $\int_\Theta \bar{q}_1(\boldsymbol{\theta})\bar{q}_2(\boldsymbol{\theta})d\boldsymbol{\theta}$ is small. Moreover, If we set $q_1(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\mathbf{y})$, $c_1 = Z$, $q_2(\boldsymbol{\theta}) = \bar{q}(\boldsymbol{\theta})$ and $c_2 = 1$, then the identity becomes

$$Z = \frac{\mathbb{E}_{\bar{q}}\left[\pi(\boldsymbol{\theta}|\mathbf{y})\alpha(\boldsymbol{\theta})\right]}{\mathbb{E}_{P}\left[\bar{q}(\boldsymbol{\theta})\alpha(\boldsymbol{\theta})\right]}. \tag{8.60}$$

The corresponding estimator employs samples from both $\bar{q}$ and P, i.e.,

$$\widehat{Z} = \frac{\frac{1}{N_2}\sum_{j=1}^{N_2}\alpha(\mathbf{z}_j)\pi(\mathbf{z}_j|\mathbf{y})}{\frac{1}{N_1}\sum_{i=i}^{N_1}\alpha(\boldsymbol{\theta}_i)\bar{q}(\boldsymbol{\theta}_i)}, \qquad \{\boldsymbol{\theta}_i\}_{i=1}^{N_1} \sim P(\boldsymbol{\theta}|\mathbf{y}), \;\; \{\mathbf{z}_j\}_{j=1}^{N_2} \sim \bar{q}(\boldsymbol{\theta}). \tag{8.61}$$

Figure 8.1 summarizes the connections among the Eqs. (8.49), (8.59), (8.60) and the corresponding different methods. The standard IS and RIS schemes have been described in the previous sections, whereas the corresponding *locally-restricted* versions will be introduced below.

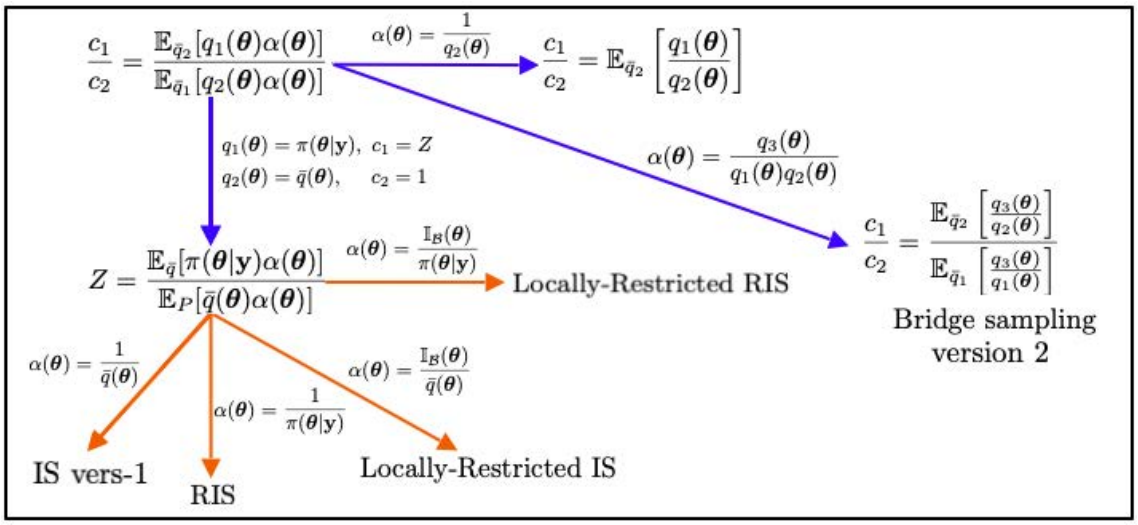


Figure 8.1: Graphical representation of the relationships among the Eqs. (8.49) (pre-umbrella identity), (8.59) (general bridge sampling identity), (8.60) (bridge sampling for Z) and the corresponding different methods, starting from bridge sampling identity (8.59).

Relationship with Chib's method

The Chib estimator, described in Section 8.3.4, is

$$\widehat{Z} = \frac{\pi(\boldsymbol{\theta}^*|\mathbf{y})\frac{1}{N_2}\sum_{j=1}^{N_2}\alpha(\boldsymbol{\theta}^*, \mathbf{v}_j)}{\frac{1}{N_1}\sum_{i=i}^{N_1}\alpha(\boldsymbol{\theta}_i, \boldsymbol{\theta}^*)\varphi(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i)}, \qquad \{\boldsymbol{\theta}_i\}_{i=1}^{N_1} \sim P(\boldsymbol{\theta}|\mathbf{y}), \;\; \{\mathbf{v}_j\}_{j=1}^{N_2} \sim \varphi(\boldsymbol{\theta}|\boldsymbol{\theta}^*), \tag{8.62}$$

where $\varphi(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$ is the proposal used inside an MCMC algorithm, $\alpha(\mathbf{x}, \mathbf{z}) : \mathbb{R}^{D_\theta} \times \mathbb{R}^{D_\theta} \to \mathbb{R}^+$ represents acceptance probability of this MCMC scheme and the point $\boldsymbol{\theta}^*$ is usually

chosen in a high probability region. Note that the balance condition involving the function φ, P and α must be satisfied,

$$\alpha(\boldsymbol{\theta}, \mathbf{z})\varphi(\mathbf{z}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\mathbf{y}) = \alpha(\mathbf{z}, \boldsymbol{\theta})\varphi(\boldsymbol{\theta}|\mathbf{z})\pi(\mathbf{z}|\mathbf{y}).$$

Using the balance condition above, if we replace $\alpha(\boldsymbol{\theta}^*, \mathbf{v}_j) = \frac{\alpha(\mathbf{v}_j, \boldsymbol{\theta}^*)\varphi(\boldsymbol{\theta}^*|\mathbf{v}_j)\pi(\mathbf{v}_j|\mathbf{y})}{\varphi(\mathbf{v}_j|\boldsymbol{\theta}^*)\pi(\boldsymbol{\theta}^*|\mathbf{y})}$ inside the numerator of (8.62), we obtain

$$\widehat{Z} = \frac{\frac{1}{N_2}\sum_{j=1}^{N_2} \frac{\varphi(\boldsymbol{\theta}^*|\mathbf{v}_j)}{\varphi(\mathbf{v}_j|\boldsymbol{\theta}^*)}\alpha(\mathbf{v}_j, \boldsymbol{\theta}^*)\pi(\mathbf{v}_j|\mathbf{y})}{\frac{1}{N_1}\sum_{i=i}^{N_1} \alpha(\boldsymbol{\theta}_i, \boldsymbol{\theta}^*)\varphi(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i)}, \tag{8.63}$$

and if we also assume a symmetric proposal $\varphi(\boldsymbol{\theta}|\boldsymbol{\theta}^*) = \varphi(\boldsymbol{\theta}^*|\boldsymbol{\theta})$, we can finally write

$$\widehat{Z} = \frac{\frac{1}{N_2}\sum_{j=1}^{N_2} \alpha(\mathbf{v}_j, \boldsymbol{\theta}^*)\pi(\mathbf{v}_j|\mathbf{y})}{\frac{1}{N_1}\sum_{i=i}^{N_1} \alpha(\boldsymbol{\theta}_i, \boldsymbol{\theta}^*)\varphi(\boldsymbol{\theta}_i|\boldsymbol{\theta}^*)}, \qquad \{\boldsymbol{\theta}_i\}_{i=1}^{N_1} \sim P(\boldsymbol{\theta}|\mathbf{y}), \;\; \{\mathbf{v}_j\}_{j=1}^{N_2} \sim \varphi(\boldsymbol{\theta}|\boldsymbol{\theta}^*), \tag{8.64}$$

We can observe a clear connection between the estimators (8.61) and (8.62). Clearly, $\varphi(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$ plays the role of $\bar{q}(\boldsymbol{\theta})$ in (8.61), and the acceptance function $\alpha(\mathbf{x}, \mathbf{z})$ plays the role of the α function in (8.61). However, in this case, $\varphi(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$ participates also *inside* the MCMC used for generating $\{\boldsymbol{\theta}_i\}_{i=1}^{N_1} \sim P(\boldsymbol{\theta}|\mathbf{y})$. The function α takes also part to the generation MCMC chain, $\{\boldsymbol{\theta}_i\}_{i=1}^{N_1}$ (being the acceptance probability of the new states), and generally its evaluation involves the evaluation of φ and P. Note also that (8.62) is more generic than (8.64), being valid also for non-symmetric proposals φ. For further discussion see [75].

Locally-restricted IS and RIS

In the literature, there exist variants of the estimators in Eqs. (8.43) and (8.46). These corrected estimators are attempts to improve the efficiency (e.g., remove the infinite variance cases, specially in the harmonic estimator) by restricting the integration to a smaller subset of Θ (usually chosen in high posterior/likelihood-valued regions) generally denoted by $\mathcal{B} \subset \Theta$. As an example, $\mathcal{B}$ can be a rectangular or ellipsoidal region centered at the *maximum a posteriori* (MAP) estimate $\widehat{\boldsymbol{\theta}}_{\text{MAP}}$.

Locally-restricted IS estimator. Consider the posterior mass of subset $\mathcal{B} \subset \Theta$,

$$Z_{\mathcal{B}} = \int_{\mathcal{B}} P(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} = \int_{\Theta} \mathbb{I}_{\mathcal{B}}(\boldsymbol{\theta})\frac{\ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})}{Z}d\boldsymbol{\theta}, \tag{8.65}$$

where $\mathbb{I}_{\mathcal{B}}(\boldsymbol{\theta})$ is an indicator function, taking value 1 for $\boldsymbol{\theta} \in \mathcal{B}$ and 0 otherwise. It leads to the following representation

$$Z = \frac{1}{Z_{\mathcal{B}}}\int_{\Theta} \mathbb{I}_{\mathcal{B}}(\boldsymbol{\theta})\ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})d\boldsymbol{\theta} = \frac{1}{Z_{\mathcal{B}}}\mathbb{E}_{\bar{q}}\left[\mathbb{I}_{\mathcal{B}}(\boldsymbol{\theta})\frac{\ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})}{\bar{q}(\boldsymbol{\theta})}\right]. \tag{8.66}$$

We can estimate $Z_{\mathcal{B}}$ considering N_1 samples from $P(\boldsymbol{\theta}|\mathbf{y})$ by taking the proportion of samples inside $\mathcal{B}$. The resulting locally-restricted IS estimator of Z is

$$\widehat{Z} = \frac{\frac{1}{N_1}\sum_{i=1}^{N_1} \frac{\mathbb{I}_{\mathcal{B}}(\mathbf{z}_i)\ell(\mathbf{y}|\mathbf{z}_i)g(\mathbf{z}_i)}{\bar{q}(\mathbf{z}_i)}}{\frac{1}{N_2}\sum_{i=1}^{N_2}\mathbb{I}_{\mathcal{B}}(\boldsymbol{\theta}_i)}, \quad \{\mathbf{z}_i\}_{i=1}^{N_1} \sim \bar{q}(\boldsymbol{\theta}), \ \{\boldsymbol{\theta}_i\}_{i=1}^{N_2} \sim P(\boldsymbol{\theta}|\mathbf{y}) \quad \text{(via MCMC)}. \tag{8.67}$$

Note that the above estimator requires samples from two densities, namely the proposal $\bar{q}(\boldsymbol{\theta})$ and the posterior density $P(\boldsymbol{\theta}|\mathbf{y})$ (via MCMC).

Locally-restricted RIS estimator. To derive the locally-restricted RIS estimator, consider the mass of $\mathcal{B}$ under $\bar{q}(\boldsymbol{\theta})$,

$$\bar{Q}(\mathcal{B}) = \int_{\mathcal{B}} \bar{q}(\boldsymbol{\theta})d\boldsymbol{\theta} = Z \cdot \mathbb{E}_P\left[\mathbb{I}_{\mathcal{B}}(\boldsymbol{\theta})\frac{\bar{q}(\boldsymbol{\theta})}{\ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})}\right], \tag{8.68}$$

which leads to the following representation

$$Z = \frac{\bar{Q}(\mathcal{B})}{\mathbb{E}_P\left[\frac{\mathbb{I}_{\mathcal{B}}(\boldsymbol{\theta})\bar{q}(\boldsymbol{\theta})}{\ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})}\right]}. \tag{8.69}$$

$\bar{Q}(\mathcal{B})$ can be estimated using a sample from $\bar{q}(\boldsymbol{\theta})$ by taking the proportion of sampled values inside $\mathcal{B}$. The locally-restricted RIS estimator is

$$\widehat{Z} = \frac{\frac{1}{N_1}\sum_{i=1}^{N_1}\mathbb{I}_{\mathcal{B}}(\mathbf{z}_i)}{\frac{1}{N_2}\sum_{i=1}^{N_2}\frac{\mathbb{I}_{\mathcal{B}}(\boldsymbol{\theta}_i)\bar{q}(\boldsymbol{\theta}_i)}{\ell(\mathbf{y}|\boldsymbol{\theta}_i)g(\boldsymbol{\theta}_i)}}, \quad \{\mathbf{z}_i\}_{i=1}^{N_1} \sim \bar{q}(\boldsymbol{\theta}), \qquad \{\boldsymbol{\theta}_i\}_{i=1}^{N_2} \sim P(\boldsymbol{\theta}|\mathbf{y}). \tag{8.70}$$

Other variants, where $\mathcal{B}$ corresponds to highest density regions, can be found in [89].

Optimal construction of bridge sampling

Identities as (8.59) are associated to the bridge sampling approach. However, considering $\alpha(\boldsymbol{\theta}) = \frac{q_3(\boldsymbol{\theta})}{q_2(\boldsymbol{\theta})q_1(\boldsymbol{\theta})}$ in Eq. (8.59), bridge sampling can be also motivated from the expression

$$\frac{c_1}{c_2} = \frac{c_3/c_2}{c_3/c_1} = \frac{\mathbb{E}_{\bar{q}_2}\left[\frac{q_3(\boldsymbol{\theta})}{q_2(\boldsymbol{\theta})}\right]}{\mathbb{E}_{\bar{q}_1}\left[\frac{q_3(\boldsymbol{\theta})}{q_1(\boldsymbol{\theta})}\right]}, \tag{8.71}$$

where the density $\bar{q}_3(\boldsymbol{\theta}) \propto q_3(\boldsymbol{\theta})$ is in some sense "in between" $q_1(\boldsymbol{\theta})$ and $q_2(\boldsymbol{\theta})$. That is, instead of applying directly (8.49) to $\frac{c_1}{c_2}$, we apply it to first estimate $\frac{c_3}{c_2}$ and $\frac{c_3}{c_1}$, and then take the ratio to cancel c_3. The bridge sampling estimator of $\frac{c_1}{c_2}$ is then

$$\frac{c_1}{c_2} \approx \frac{\frac{1}{N_2}\sum_{i=1}^{N_2}\frac{q_3(\mathbf{z}_i)}{q_2(\mathbf{z}_i)}}{\frac{1}{N_1}\sum_{i=1}^{N_1}\frac{q_3(\boldsymbol{\theta}_i)}{q_1(\boldsymbol{\theta}_i)}}, \quad \{\boldsymbol{\theta}_i\}_{i=1}^{N_1} \sim \bar{q}_1(\boldsymbol{\theta}), \quad \{\mathbf{z}_i\}_{i=1}^{N_2} \sim \bar{q}_2(\boldsymbol{\theta}). \tag{8.72}$$

Remark 9. *We do not need to draw samples from $\bar{q}_3(\boldsymbol{\theta})$, but only evaluate $q_3(\boldsymbol{\theta})$. For a comparison with umbrella sampling see Table 8.7.*

Table 8.7: Joint use of three densities: comparison between bridge and umbrella sampling.

Method	$\bar{q}_1(\boldsymbol{\theta})$	$\bar{q}_3(\boldsymbol{\theta})$	$\bar{q}_2(\boldsymbol{\theta})$	Identity
Umbrella sampling	evaluate	draw from	evaluate	$\frac{c_1}{c_2} = \frac{c_1/c_3}{c_2/c_3}$ - (8.50)
Bridge sampling	draw from	evaluate	draw from	$\frac{c_1}{c_2} = \frac{c_3/c_2}{c_3/c_1}$ - (8.71)

Optimal bridge density. It can be shown that the optimal bridge density $\bar{q}_3(\boldsymbol{\theta})$ can be expressed as a weighted harmonic mean of $\bar{q}_1(\boldsymbol{\theta})$ and $\bar{q}_2(\boldsymbol{\theta})$ (with weights being the sampling rates),

$$\begin{aligned} \bar{q}_3^{\text{opt}}(\boldsymbol{\theta}) &= \frac{1}{\frac{N_2}{N_1+N_2}[\bar{q}_1(\boldsymbol{\theta})]^{-1} + \frac{N_1}{N_1+N_2}[\bar{q}_2(\boldsymbol{\theta})]^{-1}} \\ &= \frac{1}{c_2} \cdot \frac{N_1+N_2}{N_2\frac{c_1}{c_2}q_1^{-1}(\boldsymbol{\theta}) + N_1 q_2^{-1}(\boldsymbol{\theta})} \\ &\propto q_3^{\text{opt}}(\boldsymbol{\theta}) = \frac{q_1(\boldsymbol{\theta})q_2(\boldsymbol{\theta})}{N_1 q_1(\boldsymbol{\theta}) + N_2\frac{c_1}{c_2}q_2(\boldsymbol{\theta})}. \end{aligned} \tag{8.73}$$

This is an optimal bridge density if both N_i are strictly positive, $N_i > 0$, hence we draw from both $\bar{q}_i(\boldsymbol{\theta})$. Note that $\bar{q}_3^{\text{opt}}(\boldsymbol{\theta})$ depends on the unknown ratio $r = \frac{c_1}{c_2}$. Therefore, we cannot even evaluate $q_3^{\text{opt}}(\boldsymbol{\theta})$. Hence, we need to resort to the following iterative procedure to approximate the optimal bridge sampling estimator. Noting that

$$\frac{q_3^{\text{opt}}(\boldsymbol{\theta})}{q_2(\boldsymbol{\theta})} = \frac{q_1(\boldsymbol{\theta})}{N_1 q_1(\boldsymbol{\theta}) + rN_2 q_2(\boldsymbol{\theta})}, \qquad \frac{q_3^{\text{opt}}(\boldsymbol{\theta})}{q_1(\boldsymbol{\theta})} = \frac{q_2(\boldsymbol{\theta})}{N_1 q_1(\boldsymbol{\theta}) + rN_2 q_2(\boldsymbol{\theta})}. \tag{8.74}$$

The iterative procedure is formed by the following steps:

1. Start with an initial estimate $\widehat{r}^{(1)} \approx \frac{c_1}{c_2}$ (using e.g. Laplace's).
2. For $t = 1, ..., T$:
   (a) Draw $\{\boldsymbol{\theta}_i\}_{i=1}^{N_1} \sim \bar{q}_1(\boldsymbol{\theta})$ and $\{\mathbf{z}_i\}_{i=1}^{N_2} \sim \bar{q}_2(\boldsymbol{\theta})$ and iterate

$$\widehat{r}^{(t+1)} = \frac{\frac{1}{N_2}\sum_{i=1}^{N_2} \frac{q_1(\mathbf{z}_i)}{N_1 q_1(\mathbf{z}_i) + N_2 \widehat{r}^{(t)} q_2(\mathbf{z}_i)}}{\frac{1}{N_1}\sum_{i=1}^{N_1} \frac{q_2(\boldsymbol{\theta}_i)}{N_1 q_1(\boldsymbol{\theta}_i) + N_2 \widehat{r}^{(t)} q_2(\boldsymbol{\theta}_i)}}. \tag{8.75}$$

Remark 10. *In [13, Theorem 3.3], the authors show that the asymptotic error of optimal bridge sampling with $\bar{q}_3^{opt}$ in Eq.* (8.73) *is always greater than the asymptotic error of optimal umbrella sampling using $\bar{q}_3^{opt}(\boldsymbol{\theta}) \propto |\bar{q}_1(\boldsymbol{\theta}) - \bar{q}_2(\boldsymbol{\theta})|$ in Eq.* (8.51).

Optimal bridge sampling for Z. Given the considerations above, an iterative bridge sampling estimator of Z is obtained by setting $q_1(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\mathbf{y})$, $c_1 = Z$, $\bar{q}_2(\boldsymbol{\theta}) = \bar{q}(\boldsymbol{\theta})$, so that

$$\widehat{Z}^{(t+1)} = \frac{\frac{1}{N_2}\sum_{i=1}^{N_2} \frac{\pi(\mathbf{z}_i|\mathbf{y})}{N_1 \pi(\mathbf{z}_i|\mathbf{y}) + N_2 Z^{(t)} \bar{q}(\mathbf{z}_i)}}{\frac{1}{N_1}\sum_{i=1}^{N_1} \frac{\bar{q}(\boldsymbol{\theta}_i)}{N_1 \pi(\boldsymbol{\theta}_i|\mathbf{y}) + N_2 Z^{(t)} \bar{q}(\boldsymbol{\theta}_i)}}, \quad \{\mathbf{z}_i\}_{i=1}^{N_2} \sim \bar{q}(\boldsymbol{\theta}) \text{ and } \{\boldsymbol{\theta}_i\}_{i=1}^{N_1} \sim P(\boldsymbol{\theta}|\mathbf{y}). \tag{8.76}$$

for $t = 1, ..., T$. Looking at Eqs. (8.73) and (8.71), when $N_1 = 0$, that is, when all samples are drawn from $\bar{q}(\boldsymbol{\theta})$, the estimator above reduces to (non-iterative) standard IS scheme with proposal $\bar{q}(\boldsymbol{\theta})$. When $N_2 = 0$, that is, when all samples are drawn from $P(\boldsymbol{\theta}|\mathbf{y})$, the estimator becomes the (non-iterative) RIS estimator. See [13] for a comparison of optimal umbrella sampling, bridge sampling and path sampling (described in the next section). An alternative derivation of the optimal bridge sampling estimator is given in [89], by generating samples from a mixture of type $\psi(\boldsymbol{\theta}) \propto \pi(\boldsymbol{\theta}|\mathbf{y}) + \upsilon\bar{q}(\boldsymbol{\theta})$. However, the resulting estimator employs the same samples drawn from $\psi(\boldsymbol{\theta})$ in the numerator and denominator, unlike in Eq. (8.76).

Other estimators drawing from a generic proposal and the posterior

Let consider again the scenario where we have a set of samples $\{\boldsymbol{\theta}_i\}_{i=1}^{N_1}$ from the posterior $P(\boldsymbol{\theta}|\mathbf{y})$ and set $\{\mathbf{z}_i\}_{i=1}^{N_2}$ from some proposal $\bar{q}(\boldsymbol{\theta})$, as in the bridge sampling case described above. However, here we consider that these two sets $\{\widetilde{\boldsymbol{\theta}}_i\}_{i=1}^{N_1+N_2} = \{\{\boldsymbol{\theta}_i\}_{i=1}^{N_1}, \{\mathbf{z}_i\}_{i=1}^{N_2}\}$ are drawn from the mixture $\bar{q}_{\text{mix}}(\boldsymbol{\theta}) = \frac{N_1}{N_1+N_2}P(\boldsymbol{\theta}|\mathbf{y}) + \frac{N_2}{N_1+N_2}\bar{q}(\boldsymbol{\theta})$ considering a deterministic mixture sampling approach [28], Thus, we can use the IS identities that use a *single* proposal, namely Eqs. (8.49) and (8.50).

Importance sampling with mixture (M-IS). Setting $\bar{q}_1(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y})$ and $\bar{q}_2(\boldsymbol{\theta}) = \bar{q}_{\text{mix}}(\boldsymbol{\theta})$ in Eq. (8.49), we have

$$\widehat{Z}_{\text{M-IS}} = \frac{1}{N_1+N_2}\sum_{i=1}^{N_1+N_2}\frac{\pi(\widetilde{\boldsymbol{\theta}}_i|\mathbf{y})}{\bar{q}_{\text{mix}}(\widetilde{\boldsymbol{\theta}}_i)} = \frac{1}{N_1+N_2}\sum_{i=1}^{N_1+N_2}\frac{\pi(\widetilde{\boldsymbol{\theta}}_i|\mathbf{y})}{\frac{N_1}{N_1+N_2}P(\widetilde{\boldsymbol{\theta}}_i|\mathbf{y}) + \frac{N_2}{N_1+N_2}\bar{q}(\widetilde{\boldsymbol{\theta}}_i)}, \tag{8.77}$$

where $\widetilde{\boldsymbol{\theta}}_i \sim \bar{q}_{\text{mix}}(\boldsymbol{\theta}) = \frac{N_1}{N_1+N_2}P(\boldsymbol{\theta}|\mathbf{y}) + \frac{N_2}{N_1+N_2}\bar{q}(\boldsymbol{\theta})$ [28]. This estimator cannot be directly used since it requires the evaluation of $P(\boldsymbol{\theta}|\mathbf{y}) = \frac{1}{Z}\pi(\boldsymbol{\theta}|\mathbf{y})$. From an initial guess $\widetilde{Z}^{(0)}$, the following iterative procedure can be used

$$\widehat{Z}^{(t)} = \frac{1}{N_1+N_2}\sum_{i=1}^{N_1+N_2}\frac{\widehat{Z}^{(t-1)}\pi(\widetilde{\boldsymbol{\theta}}_i|\mathbf{y})}{\frac{N_1}{N_1+N_2}\pi(\widetilde{\boldsymbol{\theta}}_i|\mathbf{y}) + \frac{N_2}{N_1+N_2}\widehat{Z}^{(t-1)}\bar{q}(\widetilde{\boldsymbol{\theta}}_i)}, \quad t \in \mathbb{N}. \tag{8.78}$$

Self-IS with mixture proposal (M-Self-IS). Setting $\bar{q}_1(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y})$, $\bar{q}_2(\boldsymbol{\theta}) = \bar{q}(\boldsymbol{\theta})$ and $\bar{q}_3(\boldsymbol{\theta}) = \bar{q}_{\text{mix}}$ in Eq. (8.50), we have

$$\widehat{Z}_{\text{M-Self-IS}} = \frac{\sum_{i=1}^{N_1+N_2}\frac{\pi(\widetilde{\boldsymbol{\theta}}_i|\mathbf{y})}{\bar{q}_{\text{mix}}(\widetilde{\boldsymbol{\theta}}_i)}}{\sum_{i=1}^{N_1+N_2}\frac{\bar{q}(\widetilde{\boldsymbol{\theta}}_i)}{\bar{q}_{\text{mix}}(\widetilde{\boldsymbol{\theta}}_i)}} = \frac{\sum_{i=1}^{N_1+N_2}\frac{\pi(\widetilde{\boldsymbol{\theta}}_i|\mathbf{y})}{\frac{N_1}{N_1+N_2}P(\widetilde{\boldsymbol{\theta}}_i)+\frac{N_2}{N_1+N_2}\bar{q}(\widetilde{\boldsymbol{\theta}}_i)}}{\sum_{i=1}^{N_1+N_2}\frac{\bar{q}(\widetilde{\boldsymbol{\theta}}_i)}{\frac{N_1}{N_1+N_2}P(\widetilde{\boldsymbol{\theta}}_i|\mathbf{y})+\frac{N_2}{N_1+N_2}\bar{q}(\widetilde{\boldsymbol{\theta}}_i)}}, \tag{8.79}$$

where $\widetilde{\boldsymbol{\theta}}_i \sim \bar{q}_{\text{mix}} = \frac{N_1}{N_1+N_2}P(\boldsymbol{\theta}|\mathbf{y}) + \frac{N_2}{N_1+N_2}\bar{q}(\boldsymbol{\theta})$ (drawn in a deterministic way). As above, this estimator is not of direct use, so we need to iterate

$$\widehat{Z}^{(t)} = \frac{\sum_{i=1}^{N_1+N_2}\frac{\pi(\widetilde{\boldsymbol{\theta}}_i|\mathbf{y})}{\frac{N_1}{N_1+N_2}\pi(\widetilde{\boldsymbol{\theta}}_i)+\frac{N_2}{N_1+N_2}\widehat{Z}^{(t-1)}\bar{q}(\widetilde{\boldsymbol{\theta}}_i)}}{\sum_{i=1}^{N_1+N_2}\frac{\bar{q}(\widetilde{\boldsymbol{\theta}}_i)}{\frac{N_1}{N_1+N_2}\pi(\widetilde{\boldsymbol{\theta}}_i|\mathbf{y})+\frac{N_2}{N_1+N_2}\widehat{Z}^{(t-1)}\bar{q}(\widetilde{\boldsymbol{\theta}}_i)}}, \quad t \in \mathbb{N}. \tag{8.80}$$

This iterative estimator is very similar to the iterative optimal bridge sampling estimator in Eq. (8.76), but it uses both set of samples in numerator and denominator. This estimator is also related to the reverse logistic regression method in [36] (for more details see [13, 9], and the next section). Furthermore, the iterative estimator (8.80) is also discussed for the case $\bar{q}(\boldsymbol{\theta}) = g(\boldsymbol{\theta})$ in [79], in an attempt to exploit the advantages of the Naive Monte Carlo and the harmonic mean estimators, while removing their drawbacks.

Remark 11. *Both iterative versions* (8.78)-(8.80) *converge to the optimal bridge sampling estimator* (8.76). *See [74], for a related discussion. As we show in the simulation study, the speed of convergence of each iterative method is different. The iterative bridge sampling estimator seems to be the quickest one.*

Summary

Several techniques described in the last two subsections, including both umbrella and bridge sampling, are encompassed by the generic formula

$$\frac{c_1}{c_2} = \mathbb{E}_{\tilde{\xi}}[q_1(\boldsymbol{\theta})\alpha(\boldsymbol{\theta})] \Big/ \mathbb{E}_{\tilde{\chi}}[q_2(\boldsymbol{\theta})\alpha(\boldsymbol{\theta})] \tag{8.81}$$

as shown in Table 8.8. The techniques differ also for which densities are drawn from and which densities are just evaluated.

Table 8.8: Summary of the IS schemes (with one or two proposal pdfs), using Eq. (8.81).

$\frac{c_1}{c_2} = \mathbb{E}_{\tilde{\xi}}[q_1(\boldsymbol{\theta})\alpha(\boldsymbol{\theta})] \Big/ \mathbb{E}_{\tilde{\chi}}[q_2(\boldsymbol{\theta})\alpha(\boldsymbol{\theta})]$								
For estimating a generic ratio $c1/c2$								
Name	$\alpha(\boldsymbol{\theta})$	$\tilde{\xi}(\boldsymbol{\theta})$	$\tilde{\chi}(\boldsymbol{\theta})$	$q_1(\boldsymbol{\theta})$	$q_2(\boldsymbol{\theta})$	c_1	c_2	sampling from
Bridge Identity - Eq. (8.59)	$\alpha(\boldsymbol{\theta})$	$\bar{q}_2(\boldsymbol{\theta})$	$\bar{q}_1(\boldsymbol{\theta})$	$q_1(\boldsymbol{\theta})$	$q_2(\boldsymbol{\theta})$	c_1	c_2	$\bar{q}_1(\boldsymbol{\theta})$, $\bar{q}_2(\boldsymbol{\theta})$
Bridge Identity - Eq. (8.71)	$\frac{q_3(\boldsymbol{\theta})}{q_2(\boldsymbol{\theta})q_1(\boldsymbol{\theta})}$	$\bar{q}_2(\boldsymbol{\theta})$	$\bar{q}_1(\boldsymbol{\theta})$					$\bar{q}_1(\boldsymbol{\theta})$, $\bar{q}_2(\boldsymbol{\theta})$
Identity - Eq. (8.49)	$\frac{1}{q_2(\boldsymbol{\theta})}$	$\bar{q}_2(\boldsymbol{\theta})$	$\bar{q}_1(\boldsymbol{\theta})$					$\bar{q}_2(\boldsymbol{\theta})$
Umbrella - Eq. (8.50)	$\frac{1}{q_3(\boldsymbol{\theta})}$	$\bar{q}_3(\boldsymbol{\theta})$	$\bar{q}_3(\boldsymbol{\theta})$					$\bar{q}_3(\boldsymbol{\theta})$
For estimating Z, with one proposal								
Self-norm. IS - Eq. (8.55)	$\frac{1}{q_3(\boldsymbol{\theta})}$	$\bar{q}_3(\boldsymbol{\theta})$	$\bar{q}_3(\boldsymbol{\theta})$	$\pi(\boldsymbol{\theta}\vert\mathbf{y})$	$f(\boldsymbol{\theta})$	Z	1	$\bar{q}_3(\boldsymbol{\theta})$
IS vers-1	$1/\bar{q}(\boldsymbol{\theta})$	$\bar{q}(\boldsymbol{\theta})$	$P(\boldsymbol{\theta}\vert\mathbf{y})$	$\pi(\boldsymbol{\theta}\vert\mathbf{y})$	$\bar{q}(\boldsymbol{\theta})$			$\bar{q}(\boldsymbol{\theta})$
RIS	$1/\pi(\boldsymbol{\theta}\vert\mathbf{y})$	$\bar{q}(\boldsymbol{\theta})$	$P(\boldsymbol{\theta}\vert\mathbf{y})$	$\pi(\boldsymbol{\theta}\vert\mathbf{y})$	$\bar{q}(\boldsymbol{\theta})$			$P(\boldsymbol{\theta}\vert\mathbf{y})$
For estimating Z, with two proposals, $P(\boldsymbol{\theta}\vert\mathbf{y})$ and $\bar{q}(\boldsymbol{\theta})$								
Bridge Identity - Eq. (8.60)	$\alpha(\boldsymbol{\theta})$	$\bar{q}(\boldsymbol{\theta})$	$P(\boldsymbol{\theta}\vert\mathbf{y})$	$\pi(\boldsymbol{\theta}\vert\mathbf{y})$	$\bar{q}(\boldsymbol{\theta})$			
Locally-Restricted IS	$\mathbb{I}_{\mathcal{B}}(\boldsymbol{\theta})/\bar{q}(\boldsymbol{\theta})$	$\bar{q}(\boldsymbol{\theta})$	$P(\boldsymbol{\theta}\vert\mathbf{y})$	$\pi(\boldsymbol{\theta}\vert\mathbf{y})$	$\bar{q}(\boldsymbol{\theta})$	Z	1	$P(\boldsymbol{\theta}\vert\mathbf{y})$, $\bar{q}(\boldsymbol{\theta})$
Locally-Restricted RIS	$\mathbb{I}_{\mathcal{B}}(\boldsymbol{\theta})/\pi(\boldsymbol{\theta}\vert\mathbf{y})$	$\bar{q}(\boldsymbol{\theta})$	$P(\boldsymbol{\theta}\vert\mathbf{y})$	$\pi(\boldsymbol{\theta}\vert\mathbf{y})$	$\bar{q}(\boldsymbol{\theta})$			

8.4.3. IS based on multiple proposal densities

In this section we consider estimators of Z using samples drawn from more than two proposal densities. These schemes are usually based on the so-called tempering and/or annealing approach.

Reasons for tempering. The idea is again to consider densities that are in some sense "in the middle" between the posterior $P(\boldsymbol{\theta}|\mathbf{y})$ and an easier-to-work-with density (e.g. the prior $g(\boldsymbol{\theta})$ or some other proposal density). These densities are usually scaled version of the posterior. Generally, the scale parameter is called *temperature*.[15] For this reason, the resulting pdfs are usually named tempered posteriors and correspond to flatter, more diffuse distributions than the standard posterior. The use of the tempered pdfs usually improve the mixing of the MCMC algorithms and foster the exploration of the space Θ. Generally, it helps the Monte Carlo methods (as MCMC and IS) to find the regions of posterior high probability. The number of such middle densities is specified by the user, and in some cases, it is equivalent to the selection of a temperature schedule for linking the prior $g(\boldsymbol{\theta})$ and $P(\boldsymbol{\theta}|\mathbf{y})$. This idea is shared by the several methods, such as path sampling, power posterior methods and stepping-stone sampling described below.

First of all, we start with a general IS scheme considering different proposals $\bar{q}_n(\boldsymbol{\theta})$'s. Some of them could be tempered posteriors and the generation would be performed by an MCMC method in this case.

Multiple Importance Sampling (MIS) estimators

Here, we consider to generate samples from different proposal densities, i.e.,

$$\boldsymbol{\theta}_n \sim \bar{q}_n(\boldsymbol{\theta}), \qquad n = 1, ..., N. \tag{8.82}$$

In this scenario, different proper importance weights can be used [28, 27, 26]. The most efficient MIS scheme considers the following weights

$$w_n = \frac{\pi(\boldsymbol{\theta}_n|\mathbf{y})}{\frac{1}{N}\sum_{i=1}^{N} \bar{q}_i(\boldsymbol{\theta}_n)} = \frac{\pi(\boldsymbol{\theta}_n|\mathbf{y})}{\psi(\boldsymbol{\theta}_n)}, \tag{8.83}$$

where $\psi(\boldsymbol{\theta}_n) = \frac{1}{N}\sum_{i=1}^{N} \bar{q}_i(\boldsymbol{\theta}_n)$. Indeed, considering the set of samples $\{\boldsymbol{\theta}_n\}_{n=1}^{N}$ drawn in a deterministic order, $\boldsymbol{\theta}_n \sim \bar{q}_n(\boldsymbol{\theta})$, and given a sample $\boldsymbol{\theta}^* \in \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N\}$ uniformly chosen in

[15] The data tempering is also possible: the tempered posteriors contain less data than the complete posterior.

$\{\boldsymbol{\theta}_n\}_{n=1}^N$, then we can write $\boldsymbol{\theta}^* \sim \psi(\boldsymbol{\theta}_n)$. The standard MIS estimator is

$$\widehat{Z} = \frac{1}{N}\sum_{n=1}^N w_n = \frac{1}{N}\sum_{n=1}^N \frac{\pi(\boldsymbol{\theta}_n|\mathbf{y})}{\psi(\boldsymbol{\theta}_n)} \tag{8.84}$$

$$= \frac{1}{N}\sum_{n=1}^N \frac{g(\boldsymbol{\theta}_n)\ell(\mathbf{y}|\boldsymbol{\theta}_n)}{\psi(\boldsymbol{\theta}_n)}, \tag{8.85}$$

$$= \frac{1}{N}\sum_{n=1}^N \eta_n \ell(\mathbf{y}|\boldsymbol{\theta}_n), \qquad \boldsymbol{\theta}_n \sim \bar{q}_n(\boldsymbol{\theta}), \qquad n = 1, ..., N. \tag{8.86}$$

where $\eta_n = \frac{g(\boldsymbol{\theta}_n)}{\psi(\boldsymbol{\theta}_n)}$. The estimator is unbiased [28]. As in the standard IS scheme, an alternative biased estimator is

$$\widehat{Z} = \sum_{n=1}^N \bar{\eta}_n \ell(\mathbf{y}|\boldsymbol{\theta}_n), \qquad \boldsymbol{\theta}_n \sim \bar{q}_n(\boldsymbol{\theta}), \qquad n = 1, ..., N, \tag{8.87}$$

where $\bar{\eta}_n = \frac{\eta_n}{\sum_{i=1}^N \eta_i}$, so that $\sum_{i=1}^N \bar{\eta}_i = 1$ and we have a convex combination of likelihood values $\ell(\mathbf{y}|\boldsymbol{\theta}_n)$'s. It is a generalization of the estimator in Eq. (8.40) and recalled below in Eq. (8.88).

Tempered posteriors as proposal densities

Let recall the IS vers-2 estimator of Z in Eq. (8.40), which involves a weighted sum of likelihood evaluations at points $\{\boldsymbol{\theta}_i\}_{i=1}^N$ drawn from importance density $\bar{q}(\boldsymbol{\theta})$ (but we can evaluate only $q(\boldsymbol{\theta})) \propto \bar{q}(\boldsymbol{\theta})$),

$$\widehat{Z} = \sum_{i=1}^N \bar{\rho}_i \ell(\mathbf{y}|\boldsymbol{\theta}_i), \quad \bar{\rho}_i = \frac{\frac{g(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)}}{\sum_{n=1}^N \frac{g(\boldsymbol{\theta}_n)}{q(\boldsymbol{\theta}_n)}} \propto \frac{g(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)}, \tag{8.88}$$

where $\sum_{i=1}^N \bar{\rho}_i = 1$. Let us consider

$$\bar{q}(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y}, \beta) \propto q(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\mathbf{y}, \beta) = g(\boldsymbol{\theta})\ell(\mathbf{y}|\boldsymbol{\theta})^\beta,$$

with $\beta \in [0, 1]$. Namely, we use a tempered posterior as importance density. Note that we can evaluate only the unnormalized density $q(\boldsymbol{\theta})$. The IS estimator version 2 can be employed in this case, and we obtain $\bar{\rho}_i \propto \frac{g(\boldsymbol{\theta}_i)}{g(\boldsymbol{\theta}_i)\ell(\mathbf{y}|\boldsymbol{\theta}_i)^\beta} = \frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta}_i)^\beta}$. The resulting IS estimator version 2 is

$$\widehat{Z} = \frac{\sum_{i=1}^N \frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta}_i)^\beta}\ell(\mathbf{y}|\boldsymbol{\theta}_i)}{\sum_{i=1}^N \frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta}_i)^\beta}} \tag{8.89}$$

$$= \frac{\sum_{i=1}^N \ell(\mathbf{y}|\boldsymbol{\theta}_i)^{1-\beta}}{\sum_{i=1}^N \ell(\mathbf{y}|\boldsymbol{\theta}_i)^{-\beta}} \qquad \{\boldsymbol{\theta}_i\}_{i=1}^N \sim P(\boldsymbol{\theta}|\mathbf{y}, \beta) \quad \text{(via MCMC)}. \tag{8.90}$$

This method is denoted below as IS with a tempered posterior as proposal (IS-P). Table 8.9 shows that this technique includes different schemes for different values of β. Different possible MIS schemes can be also considered, i.e., using Eq. (8.87) for instance [28, 26].

Table 8.9: Different estimators of Z using $\bar{q}(\boldsymbol{\theta}) \propto g(\boldsymbol{\theta})\ell(\mathbf{y}|\boldsymbol{\theta})^{\beta}$ as importance density, with $\beta \in [0, 1]$.

Name	Coefficient β	Weights $\bar{\rho}_i$	Estimator $\widehat{Z} = \sum_{i=1}^{N} \bar{\rho}_i \ell(\mathbf{y}|\boldsymbol{\theta}_i)$
Naive Monte Carlo	$\beta = 0$	$\frac{1}{N}$	$\frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{y}|\boldsymbol{\theta}_i)$
Harmonic Mean Estimator	$\beta = 1$	$\frac{\frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta}_i)}}{\sum_j \frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta}_j)}}$	$\widehat{Z} = \frac{1}{\frac{1}{N}\sum_{i=1}^{N} \frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta}_i)}}$
Power posterior as proposal pdf	$0 < \beta < 1$	$\frac{\frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta}_i)^{\beta}}}{\sum_j \frac{1}{\ell(\mathbf{y}|\boldsymbol{\theta}_j)^{\beta}}}$	$\widehat{Z} = \frac{\sum_i \ell(\mathbf{y}|\boldsymbol{\theta}_i)^{1-\beta}}{\sum_i \ell(\mathbf{y}|\boldsymbol{\theta}_i)^{-\beta}}$

Remark 12. *One could consider also to draw samples from N different tempered posteriors, $\boldsymbol{\theta}_n \sim P(\boldsymbol{\theta}|\mathbf{y}, \beta_n) \propto g(\boldsymbol{\theta})\ell(\mathbf{y}|\boldsymbol{\theta})^{\beta_n}$, with $n = 1, ..., N$, and then apply deterministic mixture idea in* (8.87)*. However, in this case, we cannot evaluate properly the mixture*

$$\psi(\boldsymbol{\theta}_n) = \frac{1}{N}\sum_{i=1}^{N} P(\boldsymbol{\theta}_n|\mathbf{y}, \beta_i) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{Z(\beta_i)}\pi(\boldsymbol{\theta}_n|\mathbf{y}, \beta_i).$$

Here, the issue is a not just a global unknown normalizing constant (as usual): in this case, we do not know the weights of the mixture since all $Z(\beta) = \int_{\Theta} g(\boldsymbol{\theta})\ell(\mathbf{y}|\boldsymbol{\theta})^{\beta} d\boldsymbol{\theta}$ are unknown. This problem can be solved using the techniques described in the next sections.

Reverse logistic regression (RLR). In RLR, the idea is to apply IS with the mixture $\psi(\boldsymbol{\theta}_n)$ in the remark above. The normalizing constants $Z(\beta_i)$ are iteratively obtained by maximizing of a suitable log-likelihood, built with the samples from each tempered posterior $P(\boldsymbol{\theta}|\mathbf{y}, \beta_n)$ [36, 56, 9].

In the next section, we describe an alternative to RLR for employing different tempered posteriors as proposals.

Stepping-stone (SS) sampling

Consider again $P(\boldsymbol{\theta}|\mathbf{y}, \beta) \propto g(\boldsymbol{\theta})\ell(\mathbf{y}|\boldsymbol{\theta})^{\beta}$ and $Z(\beta) = \int_{\Theta} g(\boldsymbol{\theta})\ell(\mathbf{y}|\boldsymbol{\theta})^{\beta} d\boldsymbol{\theta}$. The goal is to estimate $Z = \frac{Z(1)}{Z(0)}$, which can be expressed as the following product, with $\beta_0 = 0$ and $\beta_K = 1$,

$$Z = \frac{Z(1)}{Z(0)} = \prod_{k=1}^{K} \frac{Z(\beta_k)}{Z(\beta_{k-1})}, \tag{8.91}$$

where β_k are often chosen as $\beta_k = \frac{k}{K}$, $k = 1, \ldots, K$, i.e., with a uniform grid in $[0, 1]$. Note that generally $Z(0) = 1$, since it is normalizing constant of the prior. The SS method is

based on the following identity,

$$\mathbb{E}_{P(\boldsymbol{\theta}|y,\beta_{k-1})}\left[\frac{\pi(\boldsymbol{\theta}|\mathbf{y},\beta_k)}{\pi(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1})}\right] = \int_\Theta \frac{\pi(\boldsymbol{\theta}|\mathbf{y},\beta_k)}{\pi(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1})} P(\boldsymbol{\theta}|y,\beta_{k-1})d\boldsymbol{\theta},$$
$$= \frac{1}{Z(\beta_{k-1})}\int_\Theta \pi(\boldsymbol{\theta}|\mathbf{y},\beta_k)d\boldsymbol{\theta} = \frac{Z(\beta_k)}{Z(\beta_{k-1})}.$$

Then, the idea of SS sampling is to estimate each ratio $r_k = \frac{Z(\beta_k)}{Z(\beta_{k-1})}$ by importance sampling as

$$r_k = \frac{Z(\beta_k)}{Z(\beta_{k-1})} = \mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1})}\left[\frac{\pi(\boldsymbol{\theta}|\mathbf{y},\beta_k)}{\pi(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1})}\right] \tag{8.92}$$

$$= \mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1})}\left[\frac{\ell(\mathbf{y}|\boldsymbol{\theta})^{\beta_k}}{\ell(\mathbf{y}|\boldsymbol{\theta})^{\beta_{k-1}}}\right] \tag{8.93}$$

$$\approx \widehat{r_k} = \frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{y}|\boldsymbol{\theta}_{i,k-1})^{\beta_k-\beta_{k-1}}, \quad \{\boldsymbol{\theta}_{i,k-1}\}_{i=1}^N \sim P(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1}). \tag{8.94}$$

Multiplying all ratio estimates yields the final estimator of Z

$$\widehat{Z} = \prod_{k=1}^{K}\widehat{r_k} = \prod_{k=1}^{K}\left(\frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{y}|\boldsymbol{\theta}_{i,k-1})^{\beta_k-\beta_{k-1}}\right), \quad \{\boldsymbol{\theta}_{i,k-1}\}_{i=1}^N \sim P(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1}). \tag{8.95}$$

For $K = 1$, we come back to the Naive MC estimator. The sampling procedure of the SS method is graphically represented in Figure 8.2.

Remark 13. *The SS estimator is unbiased, since it is a product of unbiased estimators.*

The two following methods, path sampling and power posteriors, estimate $\log Z$ instead of Z.

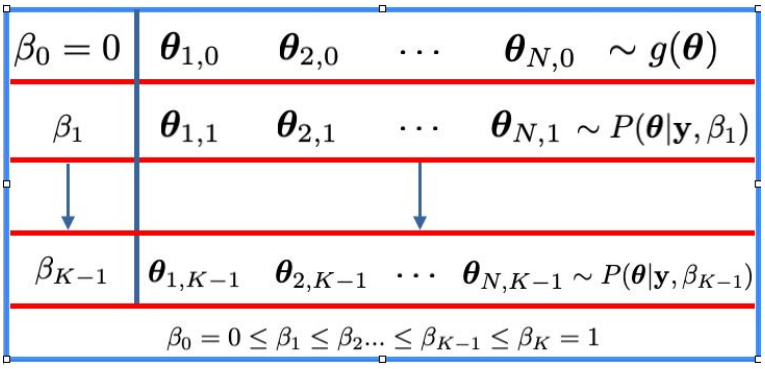


Figure 8.2: Sampling procedure in the SS method. Note that samples from $P(\boldsymbol{\theta}|\mathbf{y})$ ($\beta_K = 1$) are not considered. It is relevant to compare this figure with Figures 8.4-8.5 in the next section.

Path sampling (a.k.a., thermodynamic integration)

More specifically, the method of path sampling for estimating $\frac{c_1}{c_2}$ relies on the idea of building and drawing samples from a sequence of distributions linking $\bar{q}_1(\boldsymbol{\theta})$ and $\bar{q}_2(\boldsymbol{\theta})$ (a

continuous path). For the purpose of estimating only one constant, the marginal likelihood Z, we set $\bar{q}_2(\boldsymbol{\theta}) = g(\boldsymbol{\theta})$ and $\bar{q}_1(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y})$ and we link them by a univariate path with parameter β. Let

$$\pi(\boldsymbol{\theta}|\mathbf{y},\beta), \;\; \beta \in [0,1], \tag{8.96}$$

denote a sequence of (probably unnormalized except for $\beta = 0$) densities such $\pi(\boldsymbol{\theta}|\mathbf{y},\beta = 0) = g(\boldsymbol{\theta})$ and $\pi(\boldsymbol{\theta}|\mathbf{y},\beta = 1) = \pi(\boldsymbol{\theta}|\mathbf{y})$. More generally, we could consider $\pi(\boldsymbol{\theta}|\mathbf{y},\beta = 0) = \bar{q}(\boldsymbol{\theta})$ where $\bar{q}(\boldsymbol{\theta})$ is a generic normalized proposal density, possibly closer to the posterior than $g(\boldsymbol{\theta})$. The path sampling method for estimating the marginal likelihood is based on expressing $\log Z$ as

$$\log Z = \mathbb{E}_{p(\boldsymbol{\theta},\beta|\mathbf{y})}\left[\frac{U(\boldsymbol{\theta},\beta)}{p(\beta)}\right], \quad \text{with} \;\; U(\boldsymbol{\theta},\beta) = \frac{\partial}{\partial\beta}\log\pi(\boldsymbol{\theta}|\mathbf{y},\beta), \tag{8.97}$$

where the expectation is w.r.t. the joint $p(\boldsymbol{\theta},\beta|\mathbf{y}) = \frac{1}{Z(\beta)}\pi(\boldsymbol{\theta}|\mathbf{y},\beta)p(\beta)$, being $Z(\beta)$ the normalizing constant of $\pi(\boldsymbol{\theta}|\mathbf{y},\beta)$ and $p(\beta)$ represents a density for $\beta \in [0,1]$. Indeed, we have

$$\begin{aligned}
\mathbb{E}_{p(\boldsymbol{\theta},\beta|\mathbf{y})}\left[\frac{U(\boldsymbol{\theta},\beta)}{p(\beta)}\right] &= \int_\Theta\int_0^1 \frac{1}{p(\beta)}\left[\frac{\partial}{\partial\beta}\log\pi(\boldsymbol{\theta}|\mathbf{y},\beta)\right]\frac{\pi(\boldsymbol{\theta}|\mathbf{y},\beta)}{Z(\beta)}p(\beta)d\boldsymbol{\theta}d\beta,\\
&= \int_\Theta\int_0^1 \frac{1}{\pi(\boldsymbol{\theta}|\mathbf{y},\beta)}\left[\frac{\partial}{\partial\beta}\pi(\boldsymbol{\theta}|\mathbf{y},\beta)\right]\frac{\pi(\boldsymbol{\theta}|\mathbf{y},\beta)}{Z(\beta)}d\boldsymbol{\theta}d\beta,\\
&= \int_\Theta\int_0^1 \frac{1}{Z(\beta)}\frac{\partial}{\partial\beta}\pi(\boldsymbol{\theta}|\mathbf{y},\beta)d\boldsymbol{\theta}d\beta,\\
&= \int_0^1 \frac{1}{Z(\beta)}\frac{\partial}{\partial\beta}\left(\int_\Theta \pi(\boldsymbol{\theta}|\mathbf{y},\beta)d\boldsymbol{\theta}\right)d\beta,\\
&= \int_0^1 \frac{1}{Z(\beta)}\frac{\partial}{\partial\beta}Z(\beta)d\beta\\
&= \int_0^1 \frac{\partial}{\partial\beta}\log Z(\beta)d\beta = \log Z(1) - \log Z(0) = \log Z,
\end{aligned} \tag{8.98}$$

where we substituted $Z(\beta = 1) = Z(1) = Z$ and $Z(\beta = 0) = Z(0) = 1$. Thus, using a sample $\{\boldsymbol{\theta}_i,\beta_i\}_{i=1}^N \sim p(\boldsymbol{\theta},\beta|\mathbf{y})$, we can write the path sampling estimator for $\log Z$

$$\widehat{\log Z} = \frac{1}{N}\sum_{i=1}^N \frac{U(\boldsymbol{\theta}_i,\beta_i)}{p(\beta_i)}, \quad \{\boldsymbol{\theta}_i,\beta_i\}_{i=1}^N \sim p(\boldsymbol{\theta},\beta|\mathbf{y}). \tag{8.99}$$

The samples from $p(\boldsymbol{\theta},\beta|\mathbf{y})$ may be obtained by first drawing $\beta'(\beta_i)$ from $p(\beta)$ and then applying some MCMC steps to draw from $P(\boldsymbol{\theta}|\mathbf{y},\beta')\propto \pi(\boldsymbol{\theta}|\mathbf{y},\beta')$ given β'. Therefore, in path sampling, we have to choose (a) the path and (b) and the prior $p(\beta)$. A discussion regarding the optimal choices of the path and $p(\beta)$, see [35]. The optimal path for linking any two given densities is impractical as it depends on the normalizing constants being estimated. The geometric path described below, although suboptimal, is generic and simple to implement.

Geometric path. Often a geometric path is employed,

$$\begin{aligned}\pi(\boldsymbol{\theta}|\mathbf{y},\beta) &= g(\boldsymbol{\theta})^{1-\beta}\pi(\boldsymbol{\theta}|\mathbf{y})^{\beta} \\ &= g(\boldsymbol{\theta})\ell(\mathbf{y}|\boldsymbol{\theta})^{\beta}, \;\; \beta \in [0,1]. \end{aligned} \tag{8.100}$$

Note that $\pi(\boldsymbol{\theta}|\mathbf{y},\beta)$ is the posterior with a powered, "less informative" -"wider" likelihood (for this reason, $\pi(\boldsymbol{\theta}|\mathbf{y},\beta)$ is often called a "power posterior"). In this case, we have

$$U(\boldsymbol{\theta},\beta) = \frac{\partial}{\partial\beta}\log\pi(\boldsymbol{\theta}|\mathbf{y},\beta) = \log\ell(\mathbf{y}|\boldsymbol{\theta}),$$

so the path sampling identity becomes

$$\log Z = \mathbb{E}_{p(\boldsymbol{\theta},\beta|\mathbf{y})}\left[\frac{\log\ell(\mathbf{y}|\boldsymbol{\theta})}{p(\beta)}\right], \tag{8.101}$$

which is also used in the power posterior method of [31], described in Section 8.4.3.

Connections among path sampling, bridge sampling and stepping-stones

The path sampling method can be motivated from bridge sampling by applying the bridge sampling identity in (8.71) in a chain fashion. Assume we have $K+1$ densities $P(\boldsymbol{\theta}|\mathbf{y},\beta_k) = \pi(\boldsymbol{\theta}|\mathbf{y},\beta_k)/Z(\beta_k)$, $k = 0,\ldots,K$ from which we can draw samples, with endpoints $P(\boldsymbol{\theta}|\mathbf{y},\beta_0 = 0) = g(\boldsymbol{\theta})$ and $P(\boldsymbol{\theta}|\mathbf{y},\beta_K = 1) = P(\boldsymbol{\theta}|\mathbf{y})$. We can express $Z = Z(\beta_K) = Z(1)$ as follows

$$Z = \prod_{k=1}^{K}\frac{Z(\beta_k)}{Z(\beta_{k-1})} = \prod_{k=1}^{K}\frac{\mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1})}\left[\frac{\pi(\boldsymbol{\theta}|\mathbf{y},\beta_{k-\frac{1}{2}})}{\pi(\boldsymbol{\theta}|\mathbf{y},\beta_k)}\right]}{\mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_k)}\left[\frac{\pi(\boldsymbol{\theta}|\mathbf{y},\beta_{k-\frac{1}{2}})}{\pi(\boldsymbol{\theta}|\mathbf{y},\beta_k)}\right]}. \tag{8.102}$$

Note that we have applied the bridge sampling identity in Eq. (8.71) to each ratio $\frac{Z(\beta_k)}{Z(\beta_{k-1})}$, using $K-1$ middle densities $\pi(\boldsymbol{\theta}|\mathbf{y},\beta_{k-\frac{1}{2}})$. We can approximate the k-th term by using samples from $P(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1})$ and $P(\boldsymbol{\theta}|\mathbf{y},\beta_k)$, and take the product to obtain the final estimator of Z. Taking the logarithm of the above expression, as $K\to\infty$, results in the basic identity of path sampling for estimating Z in Eq. (8.97) [35]. In this sense, path sampling can be interpreted as a continuous application of bridge sampling steps. The difference with SS method is that it employs another identity, in (8.92), for estimating the ratios $\frac{Z(\beta_k)}{Z(\beta_{k-1})}$. Figure 8.3 summarizes the relationships among the identities (8.49)-(8.71) and their multi-stages extensions: the SS method and path sampling scheme, respectively.

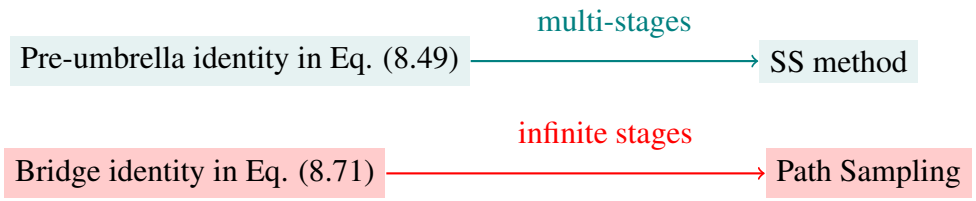


Figure 8.3: Relationships among the identities (8.49)-(8.71) and their multi-stages extensions: the SS method and path sampling scheme, respectively.

Method of Power Posteriors

The previous expression (8.101) can also be converted into an integral in $[0, 1]$ as follows

$$\begin{aligned}
\log Z &= \mathbb{E}_{p(\boldsymbol{\theta},\beta|\mathbf{y})}\left[\frac{\log \ell(\mathbf{y}|\boldsymbol{\theta})}{p(\beta)}\right], \\
&= \int_0^1 d\beta \int_\Theta \frac{\log \ell(\mathbf{y}|\boldsymbol{\theta})}{p(\beta)} \frac{\pi(\boldsymbol{\theta}|\mathbf{y},\beta)}{Z(\beta)} p(\beta) d\boldsymbol{\theta}, \\
&= \int_0^1 d\beta \int_\Theta \log \ell(\mathbf{y}|\boldsymbol{\theta}) \frac{\pi(\boldsymbol{\theta}|\mathbf{y},\beta)}{Z(\beta)} d\boldsymbol{\theta}, \\
&= \int_0^1 \mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta)}\left[\log \ell(\mathbf{y}|\boldsymbol{\theta})\right] d\beta,
\end{aligned} \tag{8.103}$$

where $P(\boldsymbol{\theta}|\mathbf{y},\beta) = \frac{\pi(\boldsymbol{\theta}|\mathbf{y},\beta)}{Z(\beta)}$ is a power posterior. The power posterior method aims at estimating the integral above by applying a quadrature rule. For instance, choosing a discretization $0 = \beta_0 < \beta_1 < \cdots < \beta_{K-1} < \beta_K = 1$, leads to approximations of order 0,

$$\widehat{\log Z} = \sum_{k=1}^{K} (\beta_k - \beta_{k-1}) \mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1})}\left[\log \ell(\mathbf{y}|\boldsymbol{\theta})\right], \tag{8.104}$$

or order 1 (trapezoidal rule),

$$\widehat{\log Z} = \sum_{k=1}^{K} (\beta_k - \beta_{k-1}) \frac{\mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_k)}\left[\log \ell(\mathbf{y}|\boldsymbol{\theta})\right] + \mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1})}\left[\log \ell(\mathbf{y}|\boldsymbol{\theta})\right]}{2}, \tag{8.105}$$

where the expected values w.r.t. the power posteriors can be independently approximated via MCMC,

$$\mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_k)}\left[\log \ell(\mathbf{y}|\boldsymbol{\theta})\right] \approx \frac{1}{N} \sum_{i=1}^{N} \log \ell(\mathbf{y}|\boldsymbol{\theta}_{i,k}), \quad \{\boldsymbol{\theta}_{i,k}\}_{i=1}^{N} \sim P(\boldsymbol{\theta}|\mathbf{y},\beta_k), \quad k = 0, \ldots, K. \tag{8.106}$$

Remark 14. *The identity* (8.103) *of method of power posteriors is derived by the path sampling identity with a geometric path, as shown in* (8.100)-(8.101)*. In this sense, the method of power posteriors is a special case of path sampling. However, unlike in path sampling, the final approximation* (8.105) *is based on a deterministic quadrature.*

Remark 15. *Note that the approximation in Eq.* (8.105) *is biased due to using a deterministic quadrature, unlike the path sampling approximation in Eq.* (8.99) *which is unbiased.*

Remark 16. *The need of using several values β_i (i.e., several tempered posteriors) seems apparent in the estimator* (8.99)-(8.105)*. For instance, in* (8.105)*, the choice a small value of K yields a poor approximation of the integral* (8.103)*. This is not the case in the SS method.*

Extensions. Several improvements of the method of power posterior have been proposed in the literature [30, 80]. In [30], the authors note that the derivative of the integrand in (8.103) corresponds to

$$\frac{d}{d\beta}\mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta)}[\log \ell(\mathbf{y}|\boldsymbol{\theta})] = \mathrm{var}_{P(\boldsymbol{\theta}|\mathbf{y},\beta)}[\log \ell(\mathbf{y}|\boldsymbol{\theta})] \tag{8.107}$$

so they propose to use this information to refine the trapezoidal rule in (8.105) by adding additional terms

$$\widehat{\log Z} = \sum_{k=1}^{K}(\beta_k - \beta_{k-1})\frac{\mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_k)}\left[\log \ell(\mathbf{y}|\boldsymbol{\theta})\right] + \mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1})}\left[\log \ell(\mathbf{y}|\boldsymbol{\theta})\right]}{2} - \tag{8.108}$$

$$\sum_{k=1}^{K}\frac{(\beta_k - \beta_{k-1})^2}{12}\left[\mathrm{var}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_k)}[\log \ell(\mathbf{y}|\boldsymbol{\theta})] - \mathrm{var}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_{k-1})}[\log \ell(\mathbf{y}|\boldsymbol{\theta})]\right], \tag{8.109}$$

This improvement comes at no extra cost since the same MCMC samples, used to estimate the expectations in (8.106), can be also used to estimate the variances in (8.109). They also propose constructing the temperature ladder recursively, starting from $\beta_0 = 0$ and $\beta_K = 1$, by leveraging the estimates of $\mathbb{E}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_k)}\left[\log \ell(\mathbf{y}|\boldsymbol{\theta})\right]$ and $\mathrm{var}_{P(\boldsymbol{\theta}|\mathbf{y},\beta_k)}[\log \ell(\mathbf{y}|\boldsymbol{\theta})]$ (for further details see [30, Sect. 2.2]). In [80], they propose the use of control variates, a variance reduction technique, in order to improve the statistical efficiency of the estimator (8.105). However, this can only be applied in settings where $\nabla_{\boldsymbol{\theta}} \log P(\boldsymbol{\theta}|\mathbf{y},\beta)$ is available.

On the selection of β_k

The method of power posteriors and SS sampling require setting an increasing sequence of β's. Some strategies for selecting the sequence of values β_k's, with $\beta_0 = 0$ and $\beta_K = 1$, are discussed, e.g., in [31, 30, 106]. A uniform sequence $\beta_k = \frac{k}{K}$ for $k = 0, \ldots, K$ can be considered, although [31] recommends putting more values near $\beta = 0$, since it is where $P(\boldsymbol{\theta}|\mathbf{y},\beta)$ is changing more rapidly. More generally, we can consider $\beta_k = (\frac{k}{K})^{1/\alpha}$. For choice of $\alpha \in [0, 1]$, the values β_k are evenly-spaced quantiles of a Beta(α,1), concentrating more and more near $\beta = 0$ as α decreases to 0 [106].

The path sampling method requires defining a prior density $p(\beta)$ from which samples are drawn. It can be shown that, for any given path, the optimal choice of $p(\beta)$ is a generalized local Jeffreys prior [35, Sect. 4.1].

Connection between stepping-stone and power posteriors methods

Taking the logarithm of the SS estimator (8.95), we obtain

$$\log \widehat{Z}_{\text{SS}} = \sum_{k=1}^{K} \log \left(\frac{1}{N} \sum_{i=1}^{N} \ell(\mathbf{y}|\boldsymbol{\theta}_{i,k-1})^{\beta_k - \beta_{k-1}} \right).$$

Applying the Jensen inequality and property of the logarithm, we can write

$$\begin{aligned} \log \widehat{Z}_{\text{SS}} &\geq \sum_{k=1}^{K} \left(\frac{1}{N} \sum_{i=1}^{N} \log \ell(\mathbf{y}|\boldsymbol{\theta}_{i,k-1})^{\beta_k - \beta_{k-1}} \right), \\ &\geq \sum_{k=1}^{K} (\beta_k - \beta_{k-1}) \left(\frac{1}{N} \sum_{i=1}^{N} \log \ell(\mathbf{y}|\boldsymbol{\theta}_{i,k-1}) \right). \end{aligned}$$

The last expression is the estimator of the power posteriors method of order 0, i.e., replacing Eq. (8.106) into (8.104). If we denote here this estimator here as $\widehat{\log Z}_{PP}$, then we have $\log \widehat{Z}_{\text{SS}} \geq \widehat{\log Z}_{PP}$. Recall also the SS estimator is unbiased.

8.5. Advanced schemes combining MCMC and IS

In the previous sections, we have already introduced several methods which require the use of MCMC algorithms in order to draw from complex proposal densities. The RIS estimator, path sampling, power posteriors and the SS sampling schemes are some examples. All these previous schemes could be assigned to the family of "MCMC-within-IS" techniques. In this section, we describe more sophisticated schemes for estimating the evidence, which combine MCMC and IS techniques: Annealed Importance Sampling (An-IS) in Section 8.5.1, Sequential Monte Carlo (SMC) in Section 8.5.2, Multiple Try Metropolis (MTM) in Section 8.5.3, and Layered Adaptive importance Sampling (LAIS) in Section 8.5.4. An-IS and SMC can be also considered "MCMC-within-IS" techniques. They provide alternative ways to employ tempered posteriors and are related to SS method, described in the previous section. We also discuss the use of MCMC transitions and resampling steps for design efficient AIS schemes. The MTM algorithm described here is an MCMC method, which belongs to the family of "IS-within-MCMC" techniques. Indeed, internal IS steps are used for proposing good candidates as new state of the chain. LAIS is an AIS scheme driven by MCMC transitions. Since the the adaptation and sampling parts can be completely separated, LAIS can be considered as a "IS-after-MCMC" technique.

8.5.1. MCMC-within-IS: weighted samples after MCMC iterations

In this section, we will see how to *properly* weight samples obtained by different MCMC iterations. We denote as $K(\mathbf{z}|\boldsymbol{\theta})$ the transition kernel which summarizes all the steps of the employed MCMC algorithm. Note that generally $K(\mathbf{z}|\boldsymbol{\theta})$ cannot be evaluated. However,

we can use MCMC kernels $K(\mathbf{z}|\boldsymbol{\theta})$ in the same fashion as proposal densities, considering the concept of the so-called *proper weighting* [54, 63].

Weighting a sample after one MCMC iteration

Let us consider the following procedure:

1. Draw $\boldsymbol{\theta}_0 \sim q(\boldsymbol{\theta})$ (where $q(\boldsymbol{\theta})$ is normalized, for simplicity).
2. Draw $\boldsymbol{\theta}_1 \sim K(\boldsymbol{\theta}_1|\boldsymbol{\theta}_0)$, where the kernel K leaves invariant density $\bar{\eta}(\boldsymbol{\theta}) = \frac{1}{c}\eta(\boldsymbol{\theta})$, i.e.,
$$\int_\Theta K(\boldsymbol{\theta}'|\boldsymbol{\theta})\bar{\eta}(\boldsymbol{\theta})d\boldsymbol{\theta} = \bar{\eta}(\boldsymbol{\theta}'). \tag{8.110}$$
3. Assign to $\boldsymbol{\theta}_1$ the weight
$$\rho(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1) = \frac{\eta(\boldsymbol{\theta}_0)}{q(\boldsymbol{\theta}_0)}\frac{\pi(\boldsymbol{\theta}_1|\mathbf{y})}{\eta(\boldsymbol{\theta}_1)}. \tag{8.111}$$

This weight is *proper* in the sense that can be used for building unbiased estimator Z (or other moments $P(\boldsymbol{\theta}|\mathbf{y})$), as described in the Liu's definition [88, Section 14.2], [54, Section 2.5.4]. Indeed, we can write

$$\begin{aligned}
\mathbb{E}[\rho(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1)] &= \int_\Theta\int_\Theta \rho(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1)K(\boldsymbol{\theta}_1|\boldsymbol{\theta}_0)q(\boldsymbol{\theta}_0)d\boldsymbol{\theta}_0 d\boldsymbol{\theta}_1, \\
&= \int_\Theta\int_\Theta \frac{\eta(\boldsymbol{\theta}_0)}{q(\boldsymbol{\theta}_0)}\frac{\pi(\boldsymbol{\theta}_1)}{\eta(\boldsymbol{\theta}_1)}K(\boldsymbol{\theta}_1|\boldsymbol{\theta}_0)q(\boldsymbol{\theta}_0)d\boldsymbol{\theta}_0 d\boldsymbol{\theta}_1, \\
&= \int_\Theta \frac{\pi(\boldsymbol{\theta}_1)}{\eta(\boldsymbol{\theta}_1)}\left[\int_\Theta \eta(\boldsymbol{\theta}_0)K(\boldsymbol{\theta}_1|\boldsymbol{\theta}_0)d\boldsymbol{\theta}_0\right]d\boldsymbol{\theta}_1, \\
&= \int_\Theta \frac{\pi(\boldsymbol{\theta}_1)}{c\bar{\eta}(\boldsymbol{\theta}_1)}c\bar{\eta}(\boldsymbol{\theta}_1)d\boldsymbol{\theta}_1 = \int_\Theta \pi(\boldsymbol{\theta}_1|\mathbf{y})d\boldsymbol{\theta}_1 = Z.
\end{aligned} \tag{8.112}$$

Note that if $\eta(\boldsymbol{\theta}) \equiv \pi(\boldsymbol{\theta}|\mathbf{y})$ then $\rho(\boldsymbol{\theta}_1) = \frac{\pi(\boldsymbol{\theta}_0|\mathbf{y})}{q(\boldsymbol{\theta}_0)}$, i.e., the IS weights remain unchanged after an MCMC iteration with invariant density $\pi(\boldsymbol{\theta}|\mathbf{y})$. Hence, if we repeat the procedure above N times generating $\{\boldsymbol{\theta}_0^{(n)}, \boldsymbol{\theta}_1^{(n)}\}_{n=1}^N$, we can build the following unbiased estimator of the Z,

$$\widehat{Z} = \frac{1}{N}\sum_{n=1}^N \rho(\boldsymbol{\theta}_0^{(n)}, \boldsymbol{\theta}_1^{(n)}) = \frac{1}{N}\sum_{n=1}^N \frac{\eta(\boldsymbol{\theta}_0^{(n)})}{q(\boldsymbol{\theta}_0^{(n)})}\frac{\pi(\boldsymbol{\theta}_1^{(n)}|\mathbf{y})}{\eta(\boldsymbol{\theta}_1^{(n)})} \tag{8.113}$$

In the next section, we extend this idea where different MCMC updates are applied, each one addressing a different invariant density.

Annealed Importance Sampling (An-IS)

In the previous section, we have considered the application of one MCMC kernel $K(\boldsymbol{\theta}_1|\boldsymbol{\theta}_0)$ (that could be formed by different MCMC steps). Below, we consider the application of several MCMC kernels addressing different target pdfs, and show their consequence in the

weighting strategy. We consider again a sequence of tempered versions of the posterior, $\pi_1(\boldsymbol{\theta}|\mathbf{y}), \pi_2(\boldsymbol{\theta}|\mathbf{y}), \ldots, \pi_L(\boldsymbol{\theta}|\mathbf{y}) \equiv \pi(\boldsymbol{\theta}|\mathbf{y})$, where the L-th version, $\pi_L(\boldsymbol{\theta}|\mathbf{y})$, coincides with the target function $\pi(\boldsymbol{\theta}|\mathbf{y})$. One possibility is to considered $\pi_i(\boldsymbol{\theta}|\mathbf{y}) = [\pi(\boldsymbol{\theta}|\mathbf{y})]^{\beta_i} = g(\boldsymbol{\theta})^{\beta_i} \ell(\mathbf{y}|\boldsymbol{\theta})^{\beta_i}$ or tempered posteriors,

$$\pi_i(\boldsymbol{\theta}|\mathbf{y}) = g(\boldsymbol{\theta})\ell(\mathbf{y}|\boldsymbol{\theta})^{\beta_i} \quad \text{where} \quad 0 \leq \beta_1 \leq \beta_2 \leq \ldots \leq \beta_L = 1. \tag{8.114}$$

as in path sampling and power posteriors. In any case, smaller β values correspond to flatter distributions.[16] The use of the tempered sequence of target pdfs usually improve the mixing of the algorithm and foster the exploration of the space Θ. Since only the last function is the true target, $\pi_L(\boldsymbol{\theta}|\mathbf{y}) = \pi(\boldsymbol{\theta}|\mathbf{y})$, different schemes have been proposed for suitable weighting the final samples.

Let us consider conditional $L-1$ kernels $K_i(\mathbf{z}|\boldsymbol{\theta})$ (with $L \geq 2$), representing the probability of different MCMC updates of jumping from the state $\boldsymbol{\theta}$ to the state $\mathbf{z}$ (note that each K_i can summarize the application of several MCMC steps), each one leaving invariant a different tempered target, $P_i(\boldsymbol{\theta}|\mathbf{y}) \propto \pi_i(\boldsymbol{\theta}|\mathbf{y})$. The Annealed Importance Sampling (An-IS) is given in Table 8.10. Note that, when $L = 2$, we have $\rho_1^{(n)} = \frac{\pi_1(\boldsymbol{\theta}_0^{(n)}|\mathbf{y})}{q(\boldsymbol{\theta}_0^{(n)})} \frac{\pi(\boldsymbol{\theta}_1^{(n)}|\mathbf{y})}{\pi_1(\boldsymbol{\theta}_1^{(n)}|\mathbf{y})}$. If,

Table 8.10: Annealed Importance Sampling (An-IS)

1. Draw N samples $\boldsymbol{\theta}_0^{(n)} \sim P_0(\boldsymbol{\theta}|\mathbf{y})$ (usually $g(\boldsymbol{\theta})$) for $n = 1, ..., N$.
2. For $k = 1, \ldots, L-1$:
   (a) Draw $\boldsymbol{\theta}_k^{(n)} \sim K_k(\boldsymbol{\theta}|\boldsymbol{\theta}_{k-1}^{(n)})$ leaving invariant $P_k(\boldsymbol{\theta}|\mathbf{y})$ for $n = 1, ..., N$, i.e., we generate N samples using an MCMC with invariant distribution $P_k(\boldsymbol{\theta}|\mathbf{y})$ (with different starting points $\boldsymbol{\theta}_{k-1}^{(n)}$).
   (b) Compute the weight associated to the sample $\boldsymbol{\theta}_k^{(n)}$, for $n = 1, ..., N$,
   $$\rho_k^{(n)} = \prod_{i=0}^{k} \frac{\pi_{i+1}(\boldsymbol{\theta}_i^{(n)}|\mathbf{y})}{\pi_i(\boldsymbol{\theta}_i^{(n)}|\mathbf{y})} = \rho_{k-1}^{(n)} \frac{\pi_{k+1}(\boldsymbol{\theta}_k^{(n)}|\mathbf{y})}{\pi_k(\boldsymbol{\theta}_k^{(n)}|\mathbf{y})}. \tag{8.115}$$
3. Return the weighted sample $\{\boldsymbol{\theta}_{L-1}^{(n)}, \rho_{L-1}^{(n)}\}_{n=1}^N$. The estimator of the marginal likelihood is
   $$\widehat{Z} = \frac{1}{N} \sum_{n=1}^{N} \rho_{L-1}^{(n)}.$$
   Combinations of An-IS with path sampling and power posterior methods can be also considered, employing the information of the rest of intermediate densities.

$\pi_1 = \pi_2 = \ldots = \pi_{L-1} = \eta \neq \pi$, then the weight is $\rho_{L-1} = \frac{\eta(\boldsymbol{\theta}_0^{(n)})}{P_0(\boldsymbol{\theta}_0^{(n)}|\mathbf{y})} \frac{\pi(\boldsymbol{\theta}_{L-1}^{(n)}|\mathbf{y})}{\eta(\boldsymbol{\theta}_{L-1}^{(n)})}$.

[16] Another alternative is to use the so-called *data tempering* [17], for instance, setting $\pi_i(\boldsymbol{\theta}|\mathbf{y}) \propto p(\boldsymbol{\theta}|y_1, \ldots, y_{d+i})$, where $d \geq 1$ and $d + L = D_y$ (recall that $\mathbf{y} = [y_1, \ldots, y_{D_y}] \in \mathbb{R}^{D_y}$).

The method above can be modified by incorporating an additional MCMC transition $\boldsymbol{\theta}_L \sim K_L(\boldsymbol{\theta}|\boldsymbol{\theta}_{L-1})$, which leaves invariant $P_L(\boldsymbol{\theta}|\mathbf{y}) = P(\boldsymbol{\theta}|\mathbf{y})$. However, since $P_L(\boldsymbol{\theta}|\mathbf{y})$ is the true target pdf, as we have seen above the weight remains unchanged (see the case $\bar{\eta}(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y})$ in the previous section). Hence, in this scenario, the output would be $\{\boldsymbol{\theta}_L^{(n)}, \rho_L^{(n)}\} = \{\boldsymbol{\theta}_L^{(n)}, \rho_{L-1}^{(n)}\}$, i.e., $\rho_L^{(n)} = \rho_{L-1}^{(n)}$. This method has been proposed in [78] but similarly schemes can be found in [17, 37].

Remark 17. *The stepping-stones (SS) sampling method described in Section 8.4.3 is strictly connected to an Ann-IS scheme. See Figures 8.2 and 8.4 for a comparison of the sampling procedures.*

Interpretation as Standard IS. For the sake of simplicity, here we consider *reversible* kernels, i.e., each kernel satisfies the detailed balance condition

$$\pi_i(\boldsymbol{\theta}|\mathbf{y})K_i(\mathbf{z}|\boldsymbol{\theta}) = \pi_i(\mathbf{z}|\mathbf{y})K_i(\boldsymbol{\theta}|\mathbf{z}) \quad \text{so that} \quad \frac{K_i(\mathbf{z}|\boldsymbol{\theta})}{K_i(\boldsymbol{\theta}|\mathbf{z})} = \frac{\pi_i(\mathbf{z}|\mathbf{y})}{\pi_i(\boldsymbol{\theta}|\mathbf{y})}. \tag{8.116}$$

We show that the weighting strategy suggested by An-IS can be interpreted as a standard IS weighting considering the following extended target density, defined in the extended space Θ^L,

$$\pi_g(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{L-1}|\mathbf{y}) = \pi(\boldsymbol{\theta}_{L-1}|\mathbf{y}) \prod_{k=1}^{L-1} K_k(\boldsymbol{\theta}_{k-1}|\boldsymbol{\theta}_k). \tag{8.117}$$

Note that π_g has the true target π as a marginal pdf. Let also consider an extended proposal pdf defined as

$$q_g(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{L-1}) = P_0(\boldsymbol{\theta}_0|\mathbf{y}) \prod_{k=1}^{L-1} K_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}). \tag{8.118}$$

The standard IS weight of an extended sample $[\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{L-1}]$ in the extended space Θ^L is

$$w(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{L-1}) = \frac{\pi_g(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{L-1}|\mathbf{y})}{q_g(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{L-1})} = \frac{\pi(\boldsymbol{\theta}_{L-1}|\mathbf{y}) \prod_{k=1}^{L-1} K_k(\boldsymbol{\theta}_{k-1}|\boldsymbol{\theta}_k)}{P_0(\boldsymbol{\theta}_0|\mathbf{y}) \prod_{k=1}^{L-1} K_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1})}. \tag{8.119}$$

Replacing the expression $\frac{K_i(\mathbf{z}|\boldsymbol{\theta})}{K_i(\boldsymbol{\theta}|\mathbf{z})} = \frac{\pi_i(\mathbf{z}|\mathbf{y})}{\pi_i(\boldsymbol{\theta}|\mathbf{y})}$ in (8.119), we obtain the Ann-IS weights

$$w(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{L-1}) = \frac{\pi(\boldsymbol{\theta}_{L-1}|\mathbf{y})}{P_0(\boldsymbol{\theta}_0|\mathbf{y})} \prod_{k=1}^{L-1} \frac{\pi_k(\boldsymbol{\theta}_{k-1}|\mathbf{y})}{\pi_k(\boldsymbol{\theta}_k|\mathbf{y})}, \tag{8.120}$$

$$= \frac{\pi_1(\boldsymbol{\theta}_0|\mathbf{y})}{P_0(\boldsymbol{\theta}_0|\mathbf{y})} \prod_{k=1}^{L-1} \frac{\pi_{k+1}(\boldsymbol{\theta}_k|\mathbf{y})}{\pi_k(\boldsymbol{\theta}_k|\mathbf{y})} = \prod_{k=0}^{L-1} \frac{\pi_{k+1}(\boldsymbol{\theta}_k|\mathbf{y})}{\pi_k(\boldsymbol{\theta}_k|\mathbf{y})} = \rho_{L-1}, \tag{8.121}$$

where we have used $\pi_L(\boldsymbol{\theta}|\mathbf{y}) = \pi(\boldsymbol{\theta}|\mathbf{y})$ and just rearranged the numerator. The sampling procedure in An-IS is graphically represented in Figure 8.4.

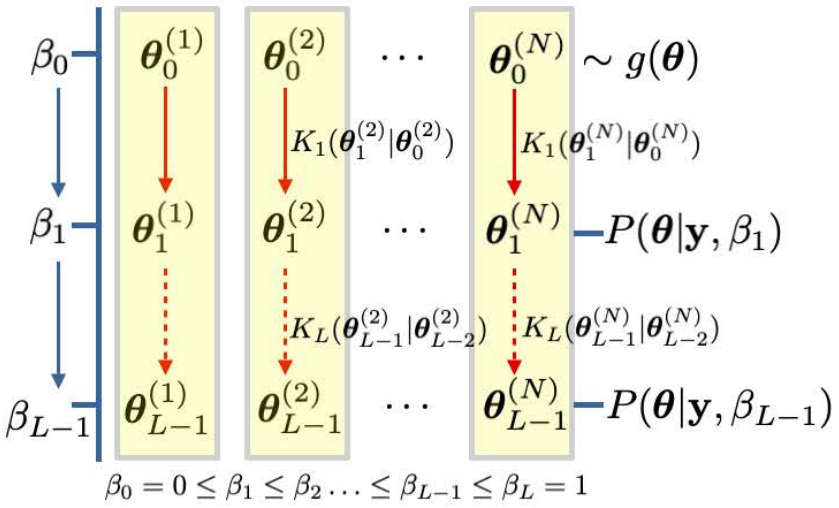


Figure 8.4: Sampling procedure in the An-IS method.

8.5.2. Weighted samples after MCMC and resampling steps

In this section, we consider also the use of resampling steps jointly with MCMC transitions. The resulting algorithm is quite sophisticated (formed by several components that should by chosen by the user) but it is a very general technique, which includes the classical particle filters, several adaptive IS (AIS) schemes and the An-IS method as special case [76].

Generic Sequential Monte Carlo

In this section, we describe a sequential IS scheme which encompasses the previous Ann-IS algorithm as a special case. The method described here uses jointly MCMC transitions and, additionally, resampling steps as well. It is called Sequential Monte Carlo (SMC), since we have a sequence of target pdfs $\pi_k(\boldsymbol{\theta}|\mathbf{y})$, $k = 1, \ldots, L$ [76]. This sequence of target densities can be defined by a state-space model as in a classical particle filtering framework (truly sequential scenario, where the goal is to track dynamic parameters). Alternatively, we can also consider a static scenario as in the previous sections, i.e., the resulting algorithm is an iterative importance sampler where we consider a sequence of *tempered* densities $\pi_k(\boldsymbol{\theta}|\mathbf{y}) = g(\boldsymbol{\theta})\ell(\mathbf{y}|\boldsymbol{\theta})^{\beta_k}$, where $0 \leq \beta_1 \leq \ldots \leq \beta_L = 1$, as in Eq.(8.114), so that $\pi_L(\boldsymbol{\theta}|\mathbf{y}) = \pi(\boldsymbol{\theta}|\mathbf{y})$ [76]. Let us again define an extended proposal density in the domain Θ^k,

$$\widetilde{q}_k(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k) = q_1(\boldsymbol{\theta}_1) \prod_{i=2}^{k} F_i(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{i-1}) : \quad \Theta^k \to \mathbb{R}, \tag{8.122}$$

where $q_1(\boldsymbol{\theta}_1)$ is a marginal proposal and $F_i(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{i-1})$ are generic forward transition pdfs, that will be used as partial proposal pdfs. Extending the space from Θ^k to Θ^{k+1} (increasing its dimension), note that we can write the recursive equation

$$\widetilde{q}_{k+1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k, \boldsymbol{\theta}_{k+1}) = F_{k+1}(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}_k)\widetilde{q}_k(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k) : \quad \Theta^{k+1} \to \mathbb{R}.$$

The marginal proposal pdfs are

$$\begin{aligned} q_k(\boldsymbol{\theta}_k) &= \int_{\Theta^{k-1}} \widetilde{q}_k(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k) d\boldsymbol{\theta}_{1:k-1} \\ &= \int_{\Theta^{k-1}} q_1(\boldsymbol{\theta}_1) \prod_{i=2}^{k} F_i(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{i-1}) d\boldsymbol{\theta}_{1:k-1}, \qquad (8.123) \\ &= \int_{\Theta} \left[\int_{\Theta^{k-2}} q_1(\boldsymbol{\theta}_1) \prod_{i=2}^{k} F_i(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{i-1}) d\boldsymbol{\theta}_{1:k-2} \right] F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}) d\boldsymbol{\theta}_{k-1}, \\ &= \int_{\Theta} q_{k-1}(\boldsymbol{\theta}_{k-1}) F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}) d\boldsymbol{\theta}_{k-1}, \qquad (8.124) \end{aligned}$$

Therefore, we would be interested in computing the *marginal* IS weights, $w_k = \frac{\pi_k(\boldsymbol{\theta}_k|\mathbf{y})}{q_k(\boldsymbol{\theta}_k)}$, for each k. However note that, in general, the marginal proposal pdfs $q_k(\boldsymbol{\theta}_k)$ cannot be computed and then cannot be evaluated. A suitable alternative approach is described next. Let us consider the extended target pdf defined as

$$\widetilde{\pi}_k(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k|\mathbf{y}) = \pi_k(\boldsymbol{\theta}_k|\mathbf{y}) \prod_{i=2}^{k} B_{i-1}(\boldsymbol{\theta}_{i-1}|\boldsymbol{\theta}_i) : \quad \Theta^k \to \mathbb{R}, \qquad (8.125)$$

$B_{i-1}(\boldsymbol{\theta}_{i-1}|\boldsymbol{\theta}_i)$ are arbitrary backward transition pdfs. Note that the space of $\{\widetilde{\pi}_k\}$ increases as k grows, and π_k is always a marginal pdf of $\widetilde{\pi}_k$. Moreover, writing the previous equation for $k+1$

$$\widetilde{\pi}_{k+1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k, \boldsymbol{\theta}_{k+1}|\mathbf{y}) = \pi_{k+1}(\boldsymbol{\theta}_{k+1}|\mathbf{y}) \prod_{i=2}^{k+1} B_{i-1}(\boldsymbol{\theta}_{i-1}|\boldsymbol{\theta}_i),$$

and writing the ratio of both, we get

$$\frac{\widetilde{\pi}_{k+1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k, \boldsymbol{\theta}_{k+1}|\mathbf{y})}{\widetilde{\pi}_k(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k|\mathbf{y})} = \frac{\pi_{k+1}(\boldsymbol{\theta}_{k+1}|\mathbf{y})}{\pi_k(\boldsymbol{\theta}_k|\mathbf{y})} B_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k+1}). \qquad (8.126)$$

Therefore, the IS weights in the extended space Θ^k are

$$\begin{aligned} w_k &= \frac{\widetilde{\pi}_k(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k|\mathbf{y})}{\widetilde{q}_k(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k)} \qquad (8.127) \\ &= \frac{\widetilde{\pi}_{k-1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{k-1}|\mathbf{y})}{\widetilde{q}_{k-1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{k-1})} \frac{\frac{\pi_k(\boldsymbol{\theta}_k|\mathbf{y})}{\pi_{k-1}(\boldsymbol{\theta}_{k-1}|\mathbf{y})} B_{k-1}(\boldsymbol{\theta}_{k-1}|\boldsymbol{\theta}_k)}{F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1})}, \qquad (8.128) \\ &= w_{k-1} \frac{\pi_k(\boldsymbol{\theta}_k|\mathbf{y}) B_{k-1}(\boldsymbol{\theta}_{k-1}|\boldsymbol{\theta}_k)}{\pi_{k-1}(\boldsymbol{\theta}_{k-1}|\mathbf{y}) F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1})}. \qquad (8.129) \end{aligned}$$

where we have replaced $w_{k-1} = \frac{\widetilde{\pi}_{k-1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{k-1}|\mathbf{y})}{\widetilde{q}_{k-1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{k-1})}$. The recursive formula in Eq. (8.129) is the key expression for several sequential IS techniques. The SMC scheme summarized in Table 8.11 is a general framework which contains different algorithms as a special cases [76]. In Table 8.11, we have used the notation $\boldsymbol{\theta}_{1:k} = [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_k]$.

Choice of the forward functions. One possible choice is to use independent proposal pdfs, i.e., $F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}) = F_k(\boldsymbol{\theta}_k)$ or random walk proposal $F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1})$, where F_k represents standard distributions (e.g., Gaussian or t-Student). An alternative is to choose $F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}) = K_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1})$, i.e., an MCMC kernel with invariant pdf $P_k(\boldsymbol{\theta}_k|\mathbf{y})$.

Table 8.11: Generic Sequential Monte Carlo (SMC)

1. Draw $\boldsymbol{\theta}_1^{(n)} \sim q_1(\boldsymbol{\theta})$, $n = 1, \ldots, N$.

2. For $k = 2, \ldots, L$:

   (a) Draw N samples $\boldsymbol{\theta}_k^{(n)} \sim F_k(\boldsymbol{\theta}|\boldsymbol{\theta}_{k-1}^{(n)})$.

   (b) Compute the weights

$$w_k^{(n)} = w_{k-1}^{(n)} \frac{\pi_k(\boldsymbol{\theta}_k^{(n)}|\mathbf{y}) B_{k-1}(\boldsymbol{\theta}_{k-1}^{(n)}|\boldsymbol{\theta}_k^{(n)})}{\pi_{k-1}(\boldsymbol{\theta}_{k-1}^{(n)}|\mathbf{y}) F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}^{(n)})}, \tag{8.130}$$

$$= w_{k-1}^{(n)} \gamma_k^{(n)}, \qquad , k = 1, \ldots, L, \tag{8.131}$$

   where we set $\gamma_k^{(n)} = \frac{\pi_k(\boldsymbol{\theta}_k^{(n)}|\mathbf{y}) B_{k-1}(\boldsymbol{\theta}_{k-1}^{(n)}|\boldsymbol{\theta}_k^{(n)})}{\pi_{k-1}(\boldsymbol{\theta}_{k-1}^{(n)}|\mathbf{y}) F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}^{(n)})}$.

   (c) Normalize the weights $\bar{w}_k^{(n)} = \frac{w_k^{(n)}}{\sum_{j=1}^N w_k^{(j)}}$, for $n = 1, ..., N$.

   (d) If $\widehat{ESS} \leq \epsilon N$:
   (with $0 \leq \epsilon \leq 1$ and $\widehat{ESS}$ is a effective sample size measure [65], see section 8.5.2)

      i. Resample N times $\{\boldsymbol{\theta}_{1:k}^{(1)}, \ldots, \boldsymbol{\theta}_{1:k}^{(N)}\}$ according to $\{\bar{w}_k^{(n)}\}_{n=1}^N$, obtaining $\{\bar{\boldsymbol{\theta}}_{1:k}^{(1)}, \ldots, \bar{\boldsymbol{\theta}}_{1:k}^{(N)}\}$.

      ii. Set $\boldsymbol{\theta}_{1:k}^{(n)} = \bar{\boldsymbol{\theta}}_{1:k}^{(n)}$, $\widehat{Z}_k = \frac{1}{N} \sum_{n=1}^N w_k^{(n)}$ and $w_k^{(n)} = \widehat{Z}_k$ for all $n = 1, \ldots, N$ [64, 63, 77, 72].

3. Return the cloud of weighted particles and

$$\widehat{Z} = \widehat{Z}_L = \frac{1}{N} \sum_{n=1}^N w_L^{(n)},$$

   if a proper weighting of the resampled particles is used (as suggested in the step 2(d)-ii above). Otherwise, you can use another estimator $\widehat{Z}_L$, as shown in Section 8.5.2 and the Supplementary Material.

Choice of backward functions. It is possible to show that the optimal backward transitions $\{B_k\}_{k=1}^L$ are [76]

$$B_{k-1}(\boldsymbol{\theta}_{k-1}|\boldsymbol{\theta}_k) = \frac{q_{k-1}(\boldsymbol{\theta}_{k-1})}{q_k(\boldsymbol{\theta}_k)} F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}). \tag{8.132}$$

This choice reduces the variance of the weights [76]. However, generally, the marginal proposal q_k in Eq. (8.123) cannot be computed (are not available), other possible $\{B_k\}$ should be considered. For instance, with the choice

$$B_{k-1}(\boldsymbol{\theta}_{k-1}|\boldsymbol{\theta}_k) = \frac{\pi_k(\boldsymbol{\theta}_{k-1}|\mathbf{y})}{\pi_k(\boldsymbol{\theta}_k|\mathbf{y})} F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}), \tag{8.133}$$

we obtain

$$w_k = w_{k-1} \frac{\pi_k(\boldsymbol{\theta}_k|\mathbf{y}) \frac{\pi_k(\boldsymbol{\theta}_{k-1}|\mathbf{y})}{\pi_k(\boldsymbol{\theta}_k|\mathbf{y})} F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1})}{\pi_{k-1}(\boldsymbol{\theta}_{k-1}|\mathbf{y}) F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1})} \tag{8.134}$$

$$= w_{k-1} \frac{\pi_k(\boldsymbol{\theta}_{k-1}|\mathbf{y})}{\pi_{k-1}(\boldsymbol{\theta}_{k-1}|\mathbf{y})}, \tag{8.135}$$

which is exactly the update rule for the weights in An-IS.

Remark 18. *With the choice of $B_{k-1}(\boldsymbol{\theta}_{k-1}|\boldsymbol{\theta}_k)$ as in Eq. 8.133, and if $F_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}) = K_k(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1})$ is an MCMC kernel with invariant $P_k(\boldsymbol{\theta}_k|\mathbf{y})$, then we come back to An-IS algorithm [78, 17, 37], described in Table 8.10. Hence, the An-IS scheme is a special case of SMC method.*

Several other methods are contained as special cases of algorithm in Table 8.11, with specific choice of $\{B_k\}$, $\{K_k\}$ and $\{\pi_k\}$, e.g., the Population Monte Carlo (PMC) method [10], that is a well-known AIS scheme. The sampling procedure in SMC is graphically represented in Figure 8.5.

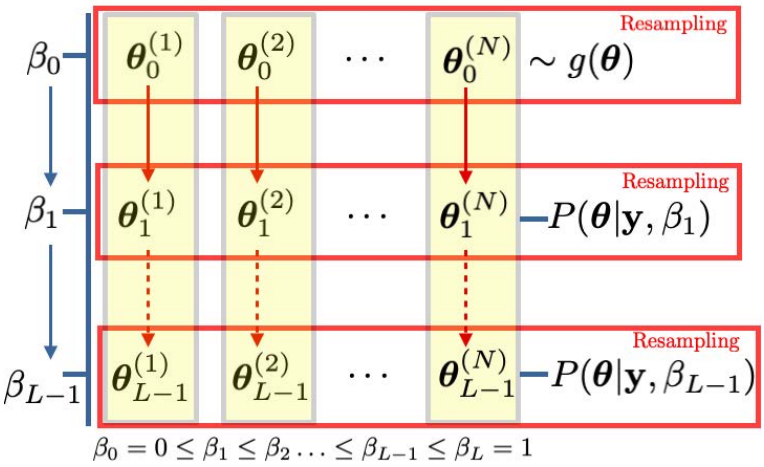


Figure 8.5: Sampling procedure in SMC. In this figure, we have considered resampling steps at each iteration ($\epsilon = 1$).

Evidence computation in a sequential framework with resampling steps

The generic algorithm in Table 8.11 employs also resampling steps. Resampling consists in drawing particles from the current cloud according to the normalized importance weights $\bar{w}_k^{(n)}$, for $n = 1, ..., N$. The resampling steps are applied only in certain iterations taking into account an ESS approximation, such as $\widehat{ESS} = \frac{1}{\sum_{n=1}^{N} (\bar{w}_k^{(n)})^2}$, or $\widehat{ESS} = \frac{1}{\max_n \bar{w}_k^{(n)}}$ [49, 65]. Generally,if $\frac{1}{N}\widehat{ESS}$ is smaller than a pre-established threshold $\epsilon \in [0, 1]$, all the particles are resampled. Thus, the condition for the adaptive resampling can be expressed as $\widehat{\text{ESS}} < \epsilon N$. When $\epsilon = 1$, the resampling is applied at each iteration [24, 25]. If $\epsilon = 0$,

no resampling steps are applied, and we have a simple sequential importance sampling (SIS) method. There are two possible estimators of Z_k in a sequential scenario:

$$\widehat{Z}_k^{(1)} = \frac{1}{N}\sum_{n=1}^{N} w_k^{(n)} = \frac{1}{N}\sum_{n=1}^{N} w_{k-1}^{(n)}\gamma_k^{(n)} = \frac{1}{N}\sum_{n=1}^{N}\left[\prod_{j=1}^{k}\gamma_j^{(n)}\right], \tag{8.136}$$

and

$$\widehat{Z}_k^{(2)} = \prod_{j=1}^{k}\left[\sum_{n=1}^{N}\bar{w}_{j-1}^{(n)}\gamma_j^{(n)}\right]. \tag{8.137}$$

These two estimators are equivalent in SIS ($\epsilon = 0$, i.e., SMC without resampling), i.e., they are the same estimator, $\widehat{Z}_k^{(1)} = \widehat{Z}_k^{(2)}$. In SMC with $\epsilon > 0$ and a proper weighting of the resampled particles, as used in Table 8.11, the two estimators are equivalent as well [64, 63, 72]. If the proper weighting of the resampled particles is not employed, $\widehat{Z}_k^{(2)}$ is the only valid option. See Table 8.12 for a summary and the Supp. Material for more details.

Table 8.12: Possible estimators of the evidence in a sequential scenario.

Scenario	**Resampling**	**Proper Weighting** [64]	$\widehat{Z}_k^{(1)}$	$\widehat{Z}_k^{(2)}$	**Equivalence**
SMC - $\epsilon = 0$ (SIS)	x	—	✓	✓	✓
SMC - $\epsilon > 0$	✓	x	x	✓	x
SMC - $\epsilon > 0$	✓	✓	✓	✓	✓

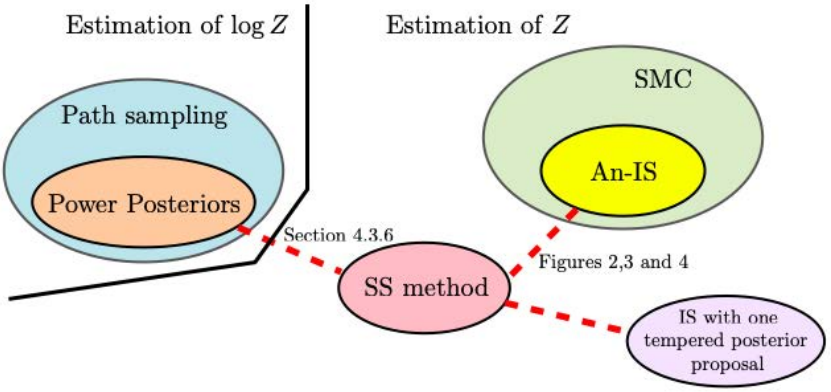


Figure 8.6: Graphical summary of the methods using tempered posteriors.

8.5.3. IS-within-MCMC: Estimation based on Multiple Try MCMC schemes

The Multiple Try Metropolis (MTM) methods are advanced MCMC algorithms which consider different candidates as possible new state of the chain [59, 70, 69]. More specifically, at each iteration different samples are generated and compared by using some proper weights. Then one of them is selected and tested as possible future state. The main advantage of these algorithms is that they foster the exploration of a larger portion of the sample space, decreasing the correlation among the states of the generated chain. Here,

we consider the use of importance weights for comparing the different candidates, in order to provide also an estimation of the marginal likelihood [70]. More specifically, we consider the Independent Multiple Try Metropolis type 2 (IMTM-2) scheme [59] with an adaptive proposal pdf. The algorithm is given in Table 8.13. The mean vector and covariance matrix are adapted using the empirical estimators yielded by all the weighted candidates drawn so far, i.e., $\{\mathbf{z}_{n,\tau}, w_{n,\tau}\}$ for all $n = 1, ..., N$ and $\tau = 1, ..., T$. Two possible estimators of the marginal likelihood can be constructed, one based on a standard adaptive importance sampling argument $\widehat{Z}^{(2)}$ [7, 8] and other based on a group importance sampling idea provided in [63].
For the sake of simplicity, we have described an independent MTM scheme, with the additional adaptation of the proposal. Random walk proposal pdfs can be also employed in an MTM algorithm [59]. In that case, the adaptation of the proposal could be not needed. However, in this scenario, the MTM algorithm requires the sampling (and weighting) of $N - 1$ additional auxiliary points. Hence, the total number of weighted samples at each iterations are $2N - 1$. These additional samples are just required for ensuring the ergodicity of the chain (including them in the acceptance probability α), but are not included as states of the Markov chain. But, for our purpose, they can be employed in the estimators of Z, as we suggest for the N candidates, $\{\mathbf{z}_{n,t}, w_{n,t}\}$, in Table 8.13. Note that the use of a random walk proposal in an MTM scheme of type in Table 8.13, could be considered as "MCMC-driven IS" method, similar to the method introduced in the next section.

8.5.4. IS-after-MCMC: Layered Adaptive Importance Sampling (LAIS)

The LAIS algorithm consider the use of N parallel (independent or interacting) MCMC chains with invariant pdf $P(\boldsymbol{\theta}|\mathbf{y})$ or a tempered version $P(\boldsymbol{\theta}|\beta)$ [67, 7]. Each MCMC chain can address a different tempered version $P(\boldsymbol{\theta}|\mathbf{y}, \beta)$ (or simply the posterior $P(\boldsymbol{\theta}|\mathbf{y})$) without jeopardizing the consistency of final estimators. After T iterations of the N MCMC schemes (upper layer), the resulting NT samples, $\{\boldsymbol{\mu}_{n,t}\}$, for $n = 1, ..., N$ and $t = 1, ..., T$ are used as location parameters of NT proposal densities $q(\boldsymbol{\theta}|\boldsymbol{\mu}_{n,t}, \mathbf{C})$. Then, these proposal pdfs are employed within a MIS scheme (lower layer), weighting the generated samples $\boldsymbol{\theta}_{n,t}$'s with the generic weight $w_{n,t} = \frac{\pi(\boldsymbol{\theta}_{n,t}|\mathbf{y})}{\Phi(\boldsymbol{\theta}_{n,t})}$ [28, 26]. In the numerator of these weights in the lower layer, we have always the unnormalized posterior $\pi(\boldsymbol{\theta}_{n,t}|\mathbf{y})$. The denominator $\Phi(\boldsymbol{\theta}_{n,t})$ is a mixture of (all or a subset of) proposal densities which specifies the type of MIS scheme applied [28, 26]. The algorithm, with different possible choices of $\Phi(\boldsymbol{\theta}_{n,t})$, is shown in Table 8.14. The first choice in (8.142) is the most costly since we have to evaluate all the proposal pdfs in all the generated samples $\boldsymbol{\theta}_{n,t}$'s, but provides the best performance in terms of efficiency of the final estimator. The second and third choices are temporal and spatial mixtures, respectively. The last choice corresponds to standard importance weights given in Section 8.4.

Let assume $P_n(\boldsymbol{\theta}|\mathbf{y}) = P(\boldsymbol{\theta}|\mathbf{y})$ for all n in the upper layer. Considering also standard parallel Metropolis-Hastings chains in the upper layer, the number of posterior evaluations in

Table 8.13: Adaptive Independent Multiple Try Metropolis type 2 (AIMTM-2)

1. Choose the initial parameters $\boldsymbol{\mu}_t$, $\mathbf{C}_t$ of the proposal q, an initial state $\boldsymbol{\theta}_0$ and a first estimation of the marginal likelihood $\widehat{Z}_0$.

2. For $t = 1, ..., T$:

   (a) Draw $\mathbf{z}_{1,t}, ..., \mathbf{z}_{N,t} \sim q(\mathbf{z}|\boldsymbol{\mu}_t, \mathbf{C}_t)$.

   (b) Compute the importance weights $w_{n,t} = \frac{\pi(\mathbf{z}_{n,t}|\mathbf{y})}{q(\mathbf{z}_{n,t}|\boldsymbol{\mu}_t, \mathbf{C}_t)}$, for $n = 1, ..., N$.

   (c) Normalize them $\bar{w}_{n,t} = \frac{w_{n,t}}{N\widehat{Z}'}$ where

   $$\widehat{Z}' = \frac{1}{N}\sum_{i=1}^{N} w_{i,t}, \qquad \text{and set} \qquad R_t = \widehat{Z}'. \tag{8.138}$$

   (d) Resample $\boldsymbol{\theta}' \in \{\mathbf{z}_{1,t}, ..., \mathbf{z}_{N,t}\}$ according to $\bar{w}_n$, with $n = 1, ..., N$.

   (e) Set $\boldsymbol{\theta}_t = \boldsymbol{\theta}'$ and $\widehat{Z}_t = \widehat{Z}'$ with probability

   $$\alpha = \min\left[1, \frac{\widehat{Z}'}{\widehat{Z}_{t-1}}\right] \tag{8.139}$$

   otherwise set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$ and $\widehat{Z}_t = \widehat{Z}_{t-1}$.

   (f) Update $\boldsymbol{\mu}_t$, $\mathbf{C}_t$ computing the corresponding empirical estimators using $\{\mathbf{z}_{n,\tau}, w_{n,\tau}\}$ for all $n = 1, ..., N$ and $\tau = 1, ..., T$.

3. Return the chain $\{\boldsymbol{\theta}_t\}_{t=1}^T$, $\{\widehat{Z}_t\}_{t=1}^T$ and $\{R_t\}_{t=1}^T$. Two possible estimators of Z can be constructed:

   $$\widehat{Z}^{(1)} = \frac{1}{T}\sum_{t=1}^{T}\widehat{Z}_t, \qquad \widehat{Z}^{(2)} = \frac{1}{T}\sum_{t=1}^{T} R_t. \tag{8.140}$$

LAIS is $2NT$. Thus, if only one chain $N = 1$ is employed in the upper layer, the number of posterior evaluations is $2T$.

Special case with recycling samples. The method in [93] can be considered as a special case of LAIS when $N = 1$, and $\{\boldsymbol{\mu}_t = \boldsymbol{\theta}_t\}$ i.e., all the samples $\{\boldsymbol{\theta}_t\}_{t=1}^T$ are generated by the unique MCMC chain with random walk proposal $\varphi(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) = q(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$ with invariant density $P(\boldsymbol{\theta}|\mathbf{y})$. In this scenario, the two layers of LAIS are collapsed in a unique layer, so that $\{\boldsymbol{\mu}_t = \boldsymbol{\theta}_t\}$. Namely, no additional generation of samples are needed in the lower layer, and the samples generated in the upper layer (via MCMC) are recycled. Hence, the number of posterior evaluations is only T. The denominator for weights used in [93] is in Eq. (8.143), i.e., a temporal mixture as in [21]. The resulting estimator is

$$\widehat{Z} = \frac{1}{T}\sum_{t=1}^{T}\frac{\pi(\boldsymbol{\theta}_t|\mathbf{y})}{\frac{1}{T}\sum_{k=1}^{T}\varphi(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1})}, \qquad \{\boldsymbol{\theta}_t\}_{t=1}^T \sim P(\boldsymbol{\theta}|\mathbf{y}) \text{ (via MCMC with a proposal } \varphi(\cdot|\cdot)).$$

Table 8.14: Layered Adaptive Importance Sampling (LAIS)

1. Generate NT samples, $\{\boldsymbol{\mu}_{n,t}\}$, using N parallel MCMC chains of length T, each MCMC method using a proposal pdf $\varphi_n(\boldsymbol{\mu}|\boldsymbol{\mu}_{t-1})$, with invariant distributions a power posterior $P_n(\boldsymbol{\theta}|\mathbf{y}) = P(\boldsymbol{\theta}|\mathbf{y},\beta_n)$ (with $\beta_n > 0$) or a posterior pdf with a smaller number of data.

2. Draw NT samples $\boldsymbol{\theta}_{n,t} \sim q(\boldsymbol{\theta}|\boldsymbol{\mu}_{n,t}, \mathbf{C})$ where $\boldsymbol{\mu}_{n,t}$ plays the role of the mean, and $\mathbf{C}$ is a covariance matrix.

3. Assign to $\boldsymbol{\theta}_{n,t}$ the weights
$$w_{n,t} = \frac{\pi(\boldsymbol{\theta}_{n,t}|\mathbf{y})}{\Phi(\boldsymbol{\theta}_{n,t})}. \tag{8.141}$$
There are different possible choices for $\Phi(\boldsymbol{\theta}_{n,t})$, for instance:
$$\Phi(\boldsymbol{\theta}_{n,t}) = \frac{1}{NT}\sum_{k=1}^{T}\sum_{i=1}^{N} q_{i,k}(\boldsymbol{\theta}_{n,t}|\boldsymbol{\mu}_{i,k}, \mathbf{C}), \tag{8.142}$$
$$\Phi(\boldsymbol{\theta}_{n,t}) = \frac{1}{T}\sum_{k=1}^{T} q(\boldsymbol{\theta}_{n,t}|\boldsymbol{\mu}_{n,k}, \mathbf{C}), \tag{8.143}$$
$$\Phi(\boldsymbol{\theta}_{n,t}) = \frac{1}{N}\sum_{i=1}^{N} q(\boldsymbol{\theta}_{n,t}|\boldsymbol{\mu}_{i,t}, \mathbf{C}), \tag{8.144}$$
$$\Phi(\boldsymbol{\theta}_{n,t}) = q(\boldsymbol{\theta}_{n,t}|\boldsymbol{\mu}_{n,t}, \mathbf{C}), \tag{8.145}$$

4. Return all the pairs $\{\boldsymbol{\theta}_{n,t}, w_{n,t}\}$, and $\widehat{Z} = \frac{1}{NT}\sum_{t=1}^{T}\sum_{n=1}^{N} w_{n,t}$.

Relationship with KDE method. LAIS can be interpreted as an extension of the KDE method in Section 8.3, where the KDE function is also employed as a proposal density in the MIS scheme. Namely, the points used in Eq. (8.26), in LAIS they are drawn from the KDE function using the deterministic mixture procedure [28, 27, 26].

Compressed LAIS (CLAIS). Let us consider the T or N is large (i.e., either large chains or several parallel chains; or both). Since NT is large, the computation of the denominators Eqs. (8.142)- (8.143)- (8.144) can be expensive. A possible solution is to use a partitioning or clustering procedure [62] with $K << NT$ clusters considering the NT samples, and then employ as denominator the function
$$\Phi(\boldsymbol{\theta}) = \sum_{k=1}^{K} \bar{a}_k \mathcal{N}(\boldsymbol{\theta}|\bar{\boldsymbol{\mu}}_k, \mathbf{C}_k), \tag{8.146}$$
where $\bar{\boldsymbol{\mu}}_k$ represents the centroid of the k-th cluster, the normalized weight $\bar{a}_k$ is proportional to the number of elements in the k-th cluster ($\sum_{k=1}^{K}\bar{a}_k = 1$), and $\mathbf{C}_k = \boldsymbol{\Sigma}_k + h\mathbf{I}$ with $\boldsymbol{\Sigma}_k$ the empirical covariance matrix of k-th cluster and $h > 0$.

Relationship with other methods using tempered posteriors. In the upper layer of

LAIS, we can use non-tempered versions of the posterior, i.e., $P_n(\boldsymbol{\theta}|\mathbf{y}) = P(\boldsymbol{\theta}|\mathbf{y})$ for all n, or tempered versions of the posterior $P_n(\boldsymbol{\theta}|\mathbf{y}) = P(\boldsymbol{\theta}|\mathbf{y},\beta_n) = \ell(\mathbf{y}|\boldsymbol{\theta})^{\beta_n} g(\boldsymbol{\theta})$. However, unlike in SS and/or power posterior methods, these samples are employed only as location parameters $\boldsymbol{\mu}_{n,t}$ of the proposal pdfs $q_{n,t}(\boldsymbol{\theta}|\boldsymbol{\mu}_{n,t}, \mathbf{C})$, and they are not included in the final estimators. Combining the tempered posteriors idea and the approach in [93], we could recycle $\boldsymbol{\theta}_{n,t} = \boldsymbol{\mu}_{n,t}$ and use $q_{n,t}(\boldsymbol{\theta}|\boldsymbol{\mu}_{n,t}) = \varphi_{n,t}(\boldsymbol{\theta}|\boldsymbol{\mu}_{n,t})$ where we denote as $\varphi_{n,t}$ the proposal pdfs employed in the MCMC chains. Another difference is that, in LAIS, the use of an "anti-tempered" posteriors with $\beta_n > 1$ is allowed and can be shown that is beneficial for the performance of the estimators (after the chains reach a good mixing) [66]. More generally, one can consider a time-varying $\beta_{n,t}$ (where t is the iteration of the n-th chain). In the first iterations, one could use $\beta_{n,t} < 1$ for fostering the exploration of the state space and helping the mixing of the chain. Then, in the last iterations, one could use $\beta_{n,t} > 1$ which increases the efficiency of the resulting IS estimators [66].

8.6. Vertical likelihood representations

In this section, we introduce a different approach based on Lebesgue representations of the integral expressing the marginal likelihood Z. First of all, we derive two one-dimensional integral representations of Z, and then we describe how it is possible to use these alternative representations by applying one-dimensional quadratures. However, the application of these quadrature rules is not straightforward. A possible final solution is the so-called nested sampling method.

8.6.1. Lebesgue representations of the marginal likelihood

First one-dimensional representation

The D_x-dimensional integral $Z = \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta}) g(\boldsymbol{\theta}) d\boldsymbol{\theta}$ can be turned into a one-dimensional integral using an extended space representation. Namely, we can write

$$Z = \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta}) g(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{8.147}$$

$$= \int_\Theta g(\boldsymbol{\theta}) d\boldsymbol{\theta} \int_0^{\ell(\mathbf{y}|\boldsymbol{\theta})} d\lambda \quad \text{(extended space representation)} \tag{8.148}$$

$$= \int_\Theta g(\boldsymbol{\theta}) d\boldsymbol{\theta} \int_0^{\infty} \mathbb{I}\{0 < \lambda < \ell(\mathbf{y}|\boldsymbol{\theta})\} d\lambda \tag{8.149}$$

where $\mathbb{I}\{0 < \lambda < \ell(\mathbf{y}|\boldsymbol{\theta})\}$ is an indicator function which is 1 if $\lambda \in [0, \ell(\mathbf{y}|\boldsymbol{\theta})]$ and 0 otherwise. Switching the integration order, we obtain

$$Z = \int_0^\infty d\lambda \int_\Theta g(\boldsymbol{\theta}) \mathbb{I}\{0 < \lambda < \ell(\mathbf{y}|\boldsymbol{\theta})\} d\boldsymbol{\theta} \tag{8.150}$$

$$= \int_0^\infty d\lambda \int_{\ell(\mathbf{y}|\boldsymbol{\theta})>\lambda} g(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{8.151}$$

$$= \int_0^\infty Z(\lambda) d\lambda = \int_0^{\sup \ell(\mathbf{y}|\boldsymbol{\theta})} Z(\lambda) d\lambda, \tag{8.152}$$

where we have set

$$Z(\lambda) = \int_{\ell(\mathbf{y}|\boldsymbol{\theta})>\lambda} g(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{8.153}$$

In Eq. (8.152), we have also assumed that $\ell(\mathbf{y}|\boldsymbol{\theta})$ is bounded so the limit of integration is $\sup \ell(\mathbf{y}|\boldsymbol{\theta})$.
Below, we define several variables and sampling procedures required for the proper understanding of the nested sampling algorithm.

The survival function $Z(\lambda)$ and related sampling procedures

The function above $Z(\lambda) : \mathbb{R}^+ \to [0, 1]$ is the mass of the prior restricted to the set $\{\boldsymbol{\theta} : \ell(\mathbf{y}|\boldsymbol{\theta}) > \lambda\}$. Note also that

$$Z(\lambda) = \mathbb{P}\left(\lambda < \ell(\mathbf{y}|\boldsymbol{\theta})\right), \quad \text{where } \boldsymbol{\theta} \sim g(\boldsymbol{\theta}). \tag{8.154}$$

Moreover, we have that $Z(\lambda) \in [0, 1]$ with $Z(0) = 1$ and $Z(\lambda') = 0$ for all $\lambda' \geq \sup \ell(\mathbf{y}|\boldsymbol{\theta})$, and it is also an non-increasing function. Therefore, $Z(\lambda)$ is a *survival function*, i.e.,

$$F(\lambda) = 1 - Z(\lambda) = \mathbb{P}\left(\ell(\mathbf{y}|\boldsymbol{\theta}) < \lambda\right) = \mathbb{P}\left(\Lambda < \lambda\right), \tag{8.155}$$

is the cumulative distribution of the random variable $\Lambda = \ell(\mathbf{y}|\boldsymbol{\theta})$ with $\boldsymbol{\theta} \sim g(\boldsymbol{\theta})$ [68, 88].

Sampling according to $F(\lambda) = 1 - Z(\lambda)$. Since $\Lambda = \ell(\mathbf{y}|\boldsymbol{\theta})$ with $\boldsymbol{\theta} \sim g(\boldsymbol{\theta})$, the following procedure generates samples λ_n from $\frac{dF(\lambda)}{d\lambda}$:

1. Draw $\boldsymbol{\theta}_n \sim g(\boldsymbol{\theta})$, for $n = 1, ..., N$.
2. Set $\lambda_n = \ell(\mathbf{y}|\boldsymbol{\theta}_n)$, , for all $n = 1, ..., N$.

Recalling the inversion method [68, Chapter 2], note also that the corresponding values

$$b_n = F(\lambda_n) \sim \mathcal{U}([0, 1]), \tag{8.156}$$

i.e., they are uniformly distributed in $[0, 1]$. Since $Z(\lambda) = 1 - F(\lambda)$, and since $V = 1 - U$ is also uniformly distributed $\mathcal{U}([0, 1])$ if $U \sim \mathcal{U}([0, 1])$, then

$$a_n = Z(\lambda_n) \sim \mathcal{U}([0, 1]). \tag{8.157}$$

In summary, finally we have that

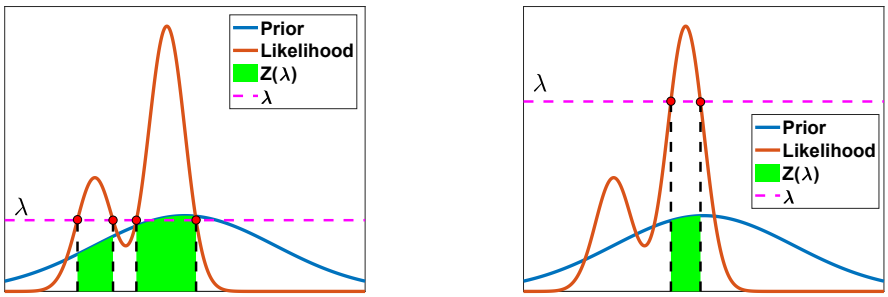
$$\text{if } \boldsymbol{\theta}_n \sim g(\boldsymbol{\theta}), \text{ and } \lambda_n = \ell(\mathbf{y}|\boldsymbol{\theta}_n) \sim F(\lambda) \quad \text{then} \quad a_n = Z(\lambda_n) \sim \mathcal{U}([0, 1]). \tag{8.158}$$

The truncated prior pdf $g(\boldsymbol{\theta}|\lambda)$ and other sampling procedures

Note that $Z(\lambda)$ is also the normalizing constant of the following truncated prior pdf

$$g(\boldsymbol{\theta}|\lambda) = \frac{1}{Z(\lambda)}\mathbb{I}\{\ell(\mathbf{y}|\boldsymbol{\theta}) > \lambda\}g(\boldsymbol{\theta}), \tag{8.159}$$

where $g(\boldsymbol{\theta}|0) = g(\boldsymbol{\theta})$ and $g(\boldsymbol{\theta}|\lambda)$ for $\lambda > 0$. Two graphical examples of $g(\boldsymbol{\theta}|\lambda)$ and $Z(\lambda)$ are given in Figure 8.7.



(a) (b)

Figure 8.7: Two examples of the area below the truncated prior $g(\boldsymbol{\theta}|\lambda)$, i.e., the function $Z(\lambda)$. Note that in figure (b) the value of λ is greater than in figure (a), so that the area $Z(\lambda)$ decreases. If λ is bigger than the maximum of the likelihood function then $Z(\lambda) = 0$.

Sampling from $g(\boldsymbol{\theta}|\lambda)$ and $F(\lambda|\lambda_0)$. Given a fixed value $\lambda_0 \geq 0$, in order to generate samples from $g(\boldsymbol{\theta}|\lambda_0)$ one alternative is to use an MCMC procedure. However, in this case, the following acceptance-rejection procedure can be also employed [68]:

1. For $n = 1, ..., N$:
   (a) Draw $\boldsymbol{\theta}' \sim g(\boldsymbol{\theta})$.
   (b) if $\ell(\mathbf{y}|\boldsymbol{\theta}') > \lambda_0$ then set $\boldsymbol{\theta}_n = \boldsymbol{\theta}'$ and $\lambda_n = \ell(\mathbf{y}|\boldsymbol{\theta}')$.
   (c) if $\ell(\mathbf{y}|\boldsymbol{\theta}') \leq \lambda_0$, then reject $\boldsymbol{\theta}'$ and repeat from step 1(a).
2. Return $\{\boldsymbol{\theta}_n\}_{n=1}^N$ and $\{\lambda_n\}_{n=1}^N$.

Observe that $\boldsymbol{\theta}_n \sim g(\boldsymbol{\theta}|\lambda_0)$, for all $n = 1, ..., N$, and the probability of accepting a generated sample $\boldsymbol{\theta}'$ is exactly $Z(\lambda)$. The values $\lambda_n = \ell(\mathbf{y}|\boldsymbol{\theta}_n)$ where $\boldsymbol{\theta}_n \sim g(\boldsymbol{\theta}|\lambda_0)$, have the following *truncated* cumulative distribution

$$F(\lambda|\lambda_0) = \frac{F(\lambda) - F(\lambda_0)}{1 - F(\lambda_0)}, \qquad \text{with } \lambda \geq \lambda_0, \tag{8.160}$$

i.e., we can write $\lambda_n \sim F(\lambda|\lambda_0)$.

Distribution of $a_n = Z(\lambda_n)$ and $\widetilde{a}_n = \frac{a_n}{a_0}$ if $\lambda_n \sim F(\lambda|\lambda_0)$

Considering the values $\lambda_n = \ell(\mathbf{y}|\boldsymbol{\theta}_n)$ where $\boldsymbol{\theta}_n \sim g(\boldsymbol{\theta}|\lambda_0)$, then $\lambda_n \sim F(\lambda|\lambda_0)$. Therefore, considering the values $a_0 = Z(\lambda_0) \leq 1$ and $a_n = Z(\lambda_n)$, with a similar argument used above in Eqs. (8.157)-(8.158) we can write

$$\begin{aligned} &a_n \sim \mathcal{U}([0, a_0]), \\ &\widetilde{a}_n = \frac{a_n}{a_0} \sim \mathcal{U}([0,1]), \quad \forall n = 1, ..., N. \end{aligned}$$

In summary, with $a_0 = Z(\lambda_0)$, we have that

$$\text{if } \boldsymbol{\theta}_n \sim g(\boldsymbol{\theta}|\lambda_0) \text{ and } \lambda_n = \ell(\mathbf{y}|\boldsymbol{\theta}_n) \sim F(\lambda|\lambda_0), \quad \text{then} \quad Z(\lambda_n) \sim \mathcal{U}([0, a_0]), \tag{8.161}$$

and the ratio $\widetilde{a}_n = \frac{a_n}{a_0} \sim \mathcal{U}([0,1])$.

Distributions $\widetilde{a}_{\texttt{max}}$

Let us consider $\lambda_1, ..., \lambda_n \sim F(\lambda|\lambda_0)$ and the minimum and maximum values

$$\lambda_{\texttt{min}} = \min_n \lambda_n, \quad a_{\texttt{max}} = Z(\lambda_{\texttt{min}}), \quad \text{and} \quad \widetilde{a}_{\texttt{max}} = \frac{a_{\texttt{max}}}{a_0} = \frac{Z(\lambda_{\texttt{min}})}{Z(\lambda_0)}. \tag{8.162}$$

Let us recall $\widetilde{a}_n = \frac{a_n}{a_0} \sim \mathcal{U}([0,1])$. Then, note that $\widetilde{a}_{\texttt{max}}$ is maximum of N uniform random variables

$$\widetilde{a}_1, ..., \widetilde{a}_N \sim \mathcal{U}([0,1]).$$

Then it is well-known that the cumulative distribution of the maximum value

$$\widetilde{a}_{\texttt{max}} = \max_n \widetilde{a}_n \sim \mathcal{B}(N, 1),$$

is distributed according to a Beta distribution $\mathcal{B}(N,1)$, i.e., $F_{\texttt{max}}(\widetilde{a}) = \widetilde{a}^N$ and density $f_{\texttt{max}}(\widetilde{a}) = \frac{dF_{\texttt{max}}(\widetilde{a})}{d\widetilde{a}} = N\widetilde{a}^{N-1}$ [68, Section 2.3.6]. In summary, we have

$$\widetilde{a}_{\texttt{max}} = \frac{Z(\lambda_{\texttt{min}})}{Z(\lambda_0)} \sim \mathcal{B}(N,1), \text{ where } \lambda_{\texttt{min}} = \min_n \lambda_n, \quad \text{and} \quad \lambda_n \sim F(\lambda|\lambda_0). \tag{8.163}$$

This result is important for deriving the standard version of the nested sampling method, described in the next section. A summary of the relationships presented above is provided in Table 8.15.

Second one-dimensional representation

Now let consider a specific *area* value $a = Z(\lambda)$. The inverse function

$$\Psi(a) = Z^{-1}(a) = \sup\{\lambda : Z(\lambda) > a\}, \tag{8.164}$$

Table 8.15: Summary of the relationships among the random variables introduced above.

Sections	Relationships
8.6.1	$Z(\lambda) = \mathbb{P}\left(\lambda < \ell(\mathbf{y}\|\boldsymbol{\theta})\right), \quad$ and $\quad F(\lambda) = 1 - Z(\lambda) = \mathbb{P}\left(\ell(\mathbf{y}\|\boldsymbol{\theta}) \leq \lambda\right), \quad$ where $\quad \boldsymbol{\theta} \sim g(\boldsymbol{\theta})$.
8.6.1	If $\boldsymbol{\theta}_n \sim g(\boldsymbol{\theta})$, we have $\lambda_n = \ell(\mathbf{y}\|\boldsymbol{\theta}_n) \sim F(\lambda)$ and $a_n = Z(\lambda_n) \sim \mathcal{U}([0,1])$.
8.6.1 8.6.1	If $\boldsymbol{\theta}_n \sim g(\boldsymbol{\theta}\|\lambda_0)$, we have $\lambda_n = \ell(\mathbf{y}\|\boldsymbol{\theta}_n) \sim F(\lambda\|\lambda_0)$ and $a_n = Z(\lambda_n) \sim \mathcal{U}([0,a_0])$, with $a_0 = Z(\lambda_0)$. Moreover, $\widetilde{a}_n = \frac{a_n}{a_0} \sim \mathcal{U}([0,1])$.
8.6.1	If $\boldsymbol{\theta}_n \sim g(\boldsymbol{\theta}\|\lambda_0)$, we have $\lambda_n = \ell(\mathbf{y}\|\boldsymbol{\theta}_n) \sim F(\lambda\|\lambda_0)$ and $\widetilde{a}_{\texttt{max}} = \frac{Z(\lambda_{\texttt{min}})}{Z(\lambda_0)} \sim \mathcal{B}(N,1)$, where $\lambda_{\texttt{min}} = \min \lambda_n$. Note also that $\widetilde{a}_{\texttt{max}} = \max \widetilde{a}_n$.

is also non-increasing. Note that $Z(\lambda) > a$ if and only if $\lambda < \Psi(a)$. Then, we can write

$$
\begin{aligned}
Z &= \int_0^\infty Z(\lambda) d\lambda \\
&= \int_0^\infty d\lambda \int_0^1 \mathbb{I}\{a < Z(\lambda)\} da \qquad \text{(again the extended space “trick”)} \\
&= \int_0^1 da \int_0^\infty \mathbb{I}\{u < Z(\lambda)\} d\lambda \qquad \text{(switching the integration order)} \\
&= \int_0^1 da \int_0^\infty \mathbb{I}\{\lambda < \Psi(a)\} d\lambda \qquad \text{(using } Z(\lambda) > a \iff \lambda < \Psi(a)\text{)} \\
&= \int_0^1 \Psi(a) da. \qquad (8.165)
\end{aligned}
$$

Summary of the one-dimensional representations

Thus, finally we have obtained two one-dimensional integrals for expressing the Bayesian evidence Z,

$$
Z = \int_0^{\sup \ell(\mathbf{y}|\boldsymbol{\theta})} Z(\lambda) d\lambda = \int_0^1 \Psi(a) da. \qquad (8.166)
$$

Now that we have expressed the quantity Z as an integral of a function over $\mathbb{R}$, we could think of applying simple quadrature: choose a grid of points in $[0, \sup \ell(\mathbf{y}|\boldsymbol{\theta})]$ ($\lambda_i > \lambda_{i-1}$)

or in $[0, 1]$ ($a_i > a_{i-1}$), evaluate $Z(\lambda)$ or $\Psi(a)$ and use the quadrature formulas

$$\widehat{Z} = \sum_{i=1}^{I} (\lambda_i - \lambda_{i-1}) Z(\lambda_i), \quad \text{or} \tag{8.167}$$

$$\widehat{Z} = \sum_{i=1}^{I} (a_i - a_{i-1}) \Psi(a_i). \tag{8.168}$$

However, this simple approach is not desirable since (i) the functions $Z(\lambda)$ and $\Psi(a)$ are intractable in most cases and (ii) they change much more rapidly over their domains than does $\pi(\boldsymbol{\theta}|\mathbf{y}) = \ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})$, hence the quadrature approximation can have very bad performance, unless the grid of points is chosen with extreme care. Table 8.16 summarizes the one-dimensional expression for $\log Z$ and Z contained in this work. Clearly, in all of them, the integrand function depends, explicitly or implicitly, on the variable $\boldsymbol{\theta}$.

Table 8.16: One-dimensional integrals for $\log Z$ and Z. Note that, in all cases, the integrand function contains the dependence on $\boldsymbol{\theta}$.

Method	Expression	Equations
path sampling	$\log Z = \int_0^1 \frac{1}{Z(\beta)} \frac{\partial}{\partial \beta} \left(\int_\Theta \pi(\boldsymbol{\theta}\|\mathbf{y}, \beta) d\boldsymbol{\theta} \right) d\beta$	Ê (8.98)
power-posteriors	$\log Z = \int_0^1 \mathbb{E}_{P(\boldsymbol{\theta}\|\mathbf{y},\beta)} [\log \ell(\mathbf{y}\|\boldsymbol{\theta})] \, d\beta$	(8.103)
vertical representation-1	$Z = \int_0^{\sup \ell(\mathbf{y}\|\boldsymbol{\theta})} Z(\lambda) d\lambda$	(8.152)-(8.153)
vertical representation-2	$Z = \int_0^1 \Psi(a) da$	(8.165)

8.6.2. Nested Sampling

Nested sampling is a technique for estimating the marginal likelihood that exploits the second identity in (8.166) [95, 18, 84]. Nested Sampling estimates Z by a quadrature using nodes (in *decreasing* order),

$$0 < a_{\texttt{max}}^{(I)} < \cdots < a_{\texttt{max}}^{(1)} < 1$$

and the quadrature formula

$$\widehat{Z} = \sum_{i=1}^{I} (a_{\texttt{max}}^{(i-1)} - a_{\texttt{max}}^{(i)}) \Psi(a_{\texttt{max}}^{(i)}) = \sum_{i=1}^{I} (a_{\texttt{max}}^{(i-1)} - a_{\texttt{max}}^{(i)}) \lambda_{\texttt{min}}^{(i)}, \tag{8.169}$$

with $a_{\texttt{max}}^{(0)} = 1$. We have to specify the grid points $a_{\texttt{max}}^{(i)}$'s (possibly well-located, with a suitable strategy) and the corresponding values $\lambda_{\texttt{min}}^{(i)} = \Psi(a_{\texttt{max}}^{(i)})$. Recall that the function $\Psi(a)$, and its inverse $a = \Psi^{-1}(\lambda) = Z(\lambda)$, are generally intractable, so that it is not even possible to evaluate $\Psi(a)$ at a grid of chosen $a_{\texttt{max}}^{(i)}$'s.

Remark 19. *The nested sampling algorithm works in the other way around: it suitably selects the ordinates $\lambda_{min}^{(i)}$'s and find some approximations $\widehat{a_i}$'s of the corresponding values $a_{max}^{(i)} = Z(\lambda_{min}^{(i)})$. This is possible since the distribution of $a_{max}^{(i)}$ is known (see Section 8.6.1).*

Choice of $\lambda_{\mathtt{min}}^{(i)}$ and $a_{\mathtt{max}}^{(i)}$ in nested sampling

Nested sampling employs an iterative procedure in order to generate an *increasing* sequence of likelihood ordinates $\lambda_{\mathtt{min}}^{(i)}$, $i = 1, ..., I$, such that

$$\lambda_{\mathtt{min}}^{(1)} < \lambda_{\mathtt{min}}^{(2)} < \lambda_{\mathtt{min}}^{(3)} .... < \lambda_{\mathtt{min}}^{(I)}. \tag{8.170}$$

The details of the algorithm is given in Table 8.17 and it is based on the sampling of the truncated prior pdf $g(\boldsymbol{\theta}|\lambda_{\mathtt{min}}^{(i-1)})$ (see Sections from 8.6.1 to 8.6.1), where i denotes the iteration index. The nested sampling procedure is explained below:

- At the first iteration ($i = 1$), we set $\lambda_{\mathtt{min}}^{(0)} = 0$ and $a_{\mathtt{max}}^{(0)} = Z(\lambda_{\mathtt{min}}^{(0)}) = 1$. Then, N samples are drawn from the prior $\boldsymbol{\theta}_n \sim g(\boldsymbol{\theta}|\lambda_{\mathtt{min}}^{(0)}) = g(\boldsymbol{\theta})$ obtaining a cloud $\mathcal{P} = \{\boldsymbol{\theta}_n\}_{n=1}^N$ and then set $\lambda_n = \ell(\mathbf{y}|\boldsymbol{\theta}_n)$, i.e., $\{\lambda_n\}_{n=1}^N \sim F(\lambda)$ as shown in Section 8.6.1. Thus, the first ordinate is chosen as

  $$\lambda_{\mathtt{min}}^{(1)} = \min_n \lambda_n = \min_n \ell(\mathbf{y}|\boldsymbol{\theta}_n) = \min_{\boldsymbol{\theta}\in\mathcal{P}} \ell(\mathbf{y}|\mathcal{P}).$$

  Since $\{\lambda_n\}_{n=1}^N \sim F(\lambda)$, using the result in Eq. (8.163), we have that

  $$\widetilde{a}_{\mathtt{max}}^{(1)} = \frac{a_{\mathtt{max}}^{(1)}}{a_{\mathtt{max}}^{(0)}} = \frac{Z(\lambda_{\mathtt{min}}^{(1)})}{Z(\lambda_{\mathtt{min}}^{(0)})} \sim \mathcal{B}(N, 1).$$

  Since $a_{\mathtt{max}}^{(0)} = Z(\lambda_{\mathtt{min}}^{(0)}) = 1$, then $\widetilde{a}_{\mathtt{max}}^{(1)} = a_{\mathtt{max}}^{(1)} \sim \mathcal{B}(N, 1)$. The corresponding $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathcal{P}} \ell(\mathbf{y}|\mathcal{P})$ is also removed from $\mathcal{P}$, i.e., $\mathcal{P} = \mathcal{P}\backslash\{\boldsymbol{\theta}^*\}$ (now $|\mathcal{P}| = N - 1$).

- At a generic i-th iteration ($i \geq 2$), a unique additional sample $\boldsymbol{\theta}'$ is drawn from the truncated prior $g(\boldsymbol{\theta}|\lambda_{\mathtt{min}}^{(i-1)})$ and added to the current cloud of samples, i.e., $\mathcal{P} = \mathcal{P} \cup \boldsymbol{\theta}'$ (now again $|\mathcal{P}| = N$). First of all, note that the value $\lambda' = \lambda_n = \ell(\mathbf{y}|\boldsymbol{\theta}')$ is distributed as $F(\lambda|\lambda_{\mathtt{min}}^{(i-1)})$ (see Section 8.6.1). More precisely, note that all the N ordinate values

  $$\{\lambda_n\}_{n=1}^N = \ell(\mathbf{y}|\mathcal{P}) = \{\lambda_n = \ell(\mathbf{y}|\boldsymbol{\theta}_n) \text{ for all } \boldsymbol{\theta}_n \in \mathcal{P}\}$$

  are distributed as $F(\lambda|\lambda_{\mathtt{min}}^{(i-1)})$, i.e., $\{\lambda_n\}_{n=1}^N \sim F(\lambda|\lambda_{\mathtt{min}}^{(i-1)})$. This is due to how the population $\mathcal{P}$ has been built in the previous iterations. Then, we choose the new minimum value as

  $$\lambda_{\mathtt{min}}^{(i)} = \min_n \lambda_n = \min_{\boldsymbol{\theta}\in\mathcal{P}} \ell(\mathbf{y}|\mathcal{P}).$$

  Moreover, since $\lambda_{\mathtt{min}}^{(i)}$ is the minimum value of $\{\lambda_1, ..., \lambda_N\} \sim F(\lambda|\lambda_{\mathtt{min}}^{(i-1)})$, in Section 8.6.1 we have seen that

  $$\widetilde{a}_{\mathtt{max}}^{(i)} = \frac{a_{\mathtt{max}}^{(i)}}{a_{\mathtt{max}}^{(i-1)}} = \frac{Z(\lambda_{\mathtt{min}}^{(i)})}{Z(\lambda_{\mathtt{min}}^{(i-1)})} \sim \mathcal{B}(N, 1), \tag{8.171}$$

where we have used Eq. (8.163). We remove again the corresponding sample $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathcal{P}} \ell(\mathbf{y}|\mathcal{P})$, i.e., we set $\mathcal{P} = \mathcal{P}\backslash\{\boldsymbol{\theta}^*\}$ and the procedure is repeated. Note that we have also found the recursion among the following random variables,

$$a_{\texttt{max}}^{(i)} = \widetilde{a}_{\texttt{max}}^{(i)} a_{\texttt{max}}^{(i-1)}, \tag{8.172}$$

for $i = 1, ..., I$ and $a_{\texttt{max}}^{(0)} = 1$.

- The random value $\widetilde{a}_{\texttt{max}}^{(i)}$ could be estimated and replaced with the expected value of the Beta distribution $\mathcal{B}(N, 1)$, i.e.,

$$\widetilde{a}_{\texttt{max}}^{(i)} \approx \widehat{a}_1 = \frac{N}{N+1} \approx \exp\left(-\frac{1}{N}\right). \tag{8.173}$$

where $\mathbb{E}[\mathcal{B}(N, 1)] = \frac{N}{N+1}$, and $\exp\left(-\frac{1}{N}\right)$ becomes a very good approximation as N grows. In that case, the recursion above becomes

$$a_{\texttt{max}}^{(i)} \approx \exp\left(-\frac{1}{N}\right) a_{\texttt{max}}^{(i-1)} = \exp\left(-\frac{i}{N}\right). \tag{8.174}$$

Then, denoting $\widehat{a}_i = \exp\left(-\frac{i}{N}\right)$, we can use $\widehat{a}_i$ as an approximation of $a_{\texttt{max}}^{(i)}$.

Remark 20. *The intuition behind the iterative approach above is to accumulate more ordinates λ_i close to the* $\sup \ell(\mathbf{y}|\boldsymbol{\theta})$. *They are also more dense around* $\sup \ell(\mathbf{y}|\boldsymbol{\theta})$. *Moreover, using this scheme, we can employ $\widehat{a}_i = \exp\left(-\frac{i}{N}\right)$ as an approximation of $a_{max}^{(i)}$.*

Remark 21. *An implicit optimization of the likelihood function is performed in the nested sampling algorithm. All population of $\lambda_i \in \mathcal{P}$ approaches the value* $\sup \ell(\mathbf{y}|\boldsymbol{\theta})$.

Further considerations

Perhaps, the most critical task of the nested sampling implementation consists in drawing from the truncated priors. For this purpose, one can use a rejection sampling or an MCMC scheme. In the first case, we sample from the prior and then accept only the samples $\boldsymbol{\theta}'$ such that $\ell(\mathbf{y}|\boldsymbol{\theta}') > \lambda$. However, as λ grows, its performance deteriorates since the acceptance probability gets smaller and smaller. The MCMC algorithms could also have poor performance due to the sample correlation, specially when the support of the constrained prior is formed by disjoint regions or distant modes [18]. Moreover, in the derivation of the standard nested sampling method we have considered different approximations. First of all, for each likelihood value λ_i, its corresponding $a_i = \Psi^{-1}(\lambda_i)$ is approximated by replacing the expected value of a Beta random variable within a recursion involving a_i (Eq. (8.172)). Then this expected value is again approximated with an exponential function in Eq. (8.173). This step could be avoided, keeping directly $\frac{N}{N+1}$. The simplicity of the final formula $\widehat{a}_i = \exp\left(-\frac{i}{N}\right)$ is perhaps the reason of using the approximation $\frac{N}{N+1} \approx \exp\left(-\frac{1}{N}\right)$. A further approximation $\mathbb{E}[a_{\texttt{max}}^{(i)}] \approx \mathbb{E}[\widetilde{a}_{\texttt{max}}^{(i)}]\mathbb{E}[a_{\texttt{max}}^{(i-1)}]$ is also implicitly applied in (8.174). Additionally, if an MCMC method is run for sampling from the constrained prior, also the likelihood values λ_i are in some sense approximated due to the possible burn-in period of the chain.

Table 8.17: The standard Nested Sampling procedure.

1. Choose N and set $\widehat{a}_0 = 1$.
2. Draw $\{\boldsymbol{\theta}_n\}_{n=1}^N \sim g(\boldsymbol{\theta})$ and define the set $\mathcal{P} = \{\boldsymbol{\theta}_n\}_{n=1}^N$. Let us also define the notation
$$\ell(\mathbf{y}|\mathcal{P}) = \{\lambda_n = \ell(\mathbf{y}|\boldsymbol{\theta}_n) \text{ for all } \boldsymbol{\theta}_n \in \mathcal{P}\}, \tag{8.175}$$
3. Set $\lambda_{\texttt{min}}^{(1)} = \min_{\boldsymbol{\theta}\in\mathcal{P}} \ell(\mathbf{y}|\mathcal{P})$ and $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathcal{P}} \ell(\mathbf{y}|\mathcal{P})$.
4. Set $\mathcal{P} = \mathcal{P}\backslash\{\boldsymbol{\theta}^*\}$, i.e., eliminate $\boldsymbol{\theta}^*$ from $\mathcal{P}$.
5. Find an approximation $\widehat{a}_1$ of $a_{\texttt{max}}^{(1)} = Z(\lambda_{\texttt{min}}^{(1)})$. One usual choice is $\widehat{a}_1 = \exp\left(-\frac{1}{N}\right)$.
6. For $i = 2, .., I$:
   (a) Draw $\boldsymbol{\theta}' \sim g(\boldsymbol{\theta}|\lambda_{\texttt{min}}^{(i-1)})$ and add to the current cloud of samples, i.e., $\mathcal{P} = \mathcal{P} \cup \boldsymbol{\theta}'$.
   (b) Set $\lambda_{\texttt{min}}^{(i)} = \min_{\boldsymbol{\theta}\in\mathcal{P}} \ell(\mathbf{y}|\mathcal{P})$ and $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathcal{P}} \ell(\mathbf{y}|\mathcal{P})$.
   (c) Set $\mathcal{P} = \mathcal{P}\backslash\{\boldsymbol{\theta}^*\}$.
   (d) Find an approximation $\widehat{a}_i$ of $a_{\texttt{max}}^{(i)} = Z(\lambda_{\texttt{min}}^{(i)})$. One usual choice is
   $$\widehat{a}_i = \exp\left(-\frac{i}{N}\right), \tag{8.176}$$
   The rationale behind this choice is explained in the section above.
7. Return
$$\widehat{Z} = \sum_{i=1}^{I} (\widehat{a}_{i-1} - \widehat{a}_i)\lambda_{\texttt{min}}^{(i)} = \sum_{i=1}^{I} (e^{-\frac{i-1}{N}} - e^{-\frac{i}{N}})\lambda_{\texttt{min}}^{(i)}. \tag{8.177}$$

Generalized Importance Sampling based on vertical representations

Let us recall the estimator IS vers-2 with proposal density $\bar{q}(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta})$,

$$\widehat{Z} = \sum_{n=1}^{N} \bar{\rho}_n \ell(\mathbf{y}|\boldsymbol{\theta}_n), \qquad \{\boldsymbol{\theta}_n\}_{n=1}^N \sim \bar{q}(\boldsymbol{\theta}), \tag{8.178}$$

where $\rho_n = \frac{g(\boldsymbol{\theta}_n)}{q(\boldsymbol{\theta}_n)}$ and $\bar{\rho}_n = \frac{\rho_n}{\sum_{n=1}^N \rho_n}$. In [84], the authors consider the use of the following proposal pdf

$$\bar{q}_w(\boldsymbol{\theta}) = \frac{g(\boldsymbol{\theta})W(\ell(\mathbf{y}|\boldsymbol{\theta}))}{Z_w} \propto q_w(\boldsymbol{\theta}) = g(\boldsymbol{\theta})W(\ell(\mathbf{y}|\boldsymbol{\theta})), \tag{8.179}$$

where the function $W(\lambda) : \mathbb{R}^+ \to \mathbb{R}^+$ is defined by the user. Using $\bar{q}_w(\boldsymbol{\theta})$ leads to the weights of the form

$$\rho_n = \frac{g(\boldsymbol{\theta}_n)}{q_w(\boldsymbol{\theta}_n)} = \frac{1}{W(\ell(\mathbf{y}|\boldsymbol{\theta}_n))}, \quad \boldsymbol{\theta}_n \sim \bar{q}_w(\boldsymbol{\theta}). \tag{8.180}$$

Note that choosing $W(\lambda) = \lambda$ we have $W(\ell(\mathbf{y}|\boldsymbol{\theta})) = \ell(\mathbf{y}|\boldsymbol{\theta})$, and $\bar{q}_w(\boldsymbol{\theta}) = P(\boldsymbol{\theta}|\mathbf{y})$, recovering the harmonic mean estimator. With $W(\lambda) = \lambda^\beta$, we have $W(\ell(\mathbf{y}|\boldsymbol{\theta})) = \ell(\mathbf{y}|\boldsymbol{\theta})^\beta$ and $\bar{q}_w(\boldsymbol{\theta}) = \frac{g(\boldsymbol{\theta})\ell(\mathbf{y}|\boldsymbol{\theta})^\beta}{Z(\beta)}$, recovering the method in Section 8.4.3 that uses a power posterior as a proposal pdf. Nested sampling seems that can be also included in this framework [84].

8.7. On the marginal likelihood approach and other strategies

In this section, we examine the marginal likelihood approach to Bayesian model selection and compare it to other strategies such as the well-known *posterior predictive check* approach.

8.7.1. Dependence on the prior and related discussion

The marginal likelihood approach for model selection and hypothesis testing naturally appears as a consequence of the application of Bayes' theorem to derive posterior model probabilities $p(\mathcal{M}_m|\mathbf{y}) \propto p_m Z_m$. Under the assumption that one of $\mathcal{M}_m$ is the true generating model, the Bayes factor will choose the correct model as the number of data grows, $D_y \to \infty$ [47]. We can also apply the posterior model probabilities $p(\mathcal{M}_m|\mathbf{y})$ to combine inferences across models, a setting called Bayesian model averaging [46, 72].

Dependence on the prior

In Section 8.2.2, we have seen the marginal likelihood Z contains intrinsically a penalization for the model complexity. This penalization is related to the choice of the prior and its "overlap" with likelihood function. Indeed, $Z = \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})d\boldsymbol{\theta}$ is by definition a continuous mixture of the likelihood values weighted according to the prior. In this sense, depending on the choice of the prior, the evidence Z can take any possible value in the interval $[\ell(\mathbf{y}|\boldsymbol{\theta}_{\min}), \ell(\mathbf{y}|\boldsymbol{\theta}_{\max})]$ (see Section 8.2.2, for more details). Hence, the marginal likelihood even with strong data (unlike the posterior density) is highly sensitivity to the choice of prior density. See also the examples in the Supplementary Material.

Improper priors. The use of improper priors, $\int_\Theta g(\boldsymbol{\theta})d\boldsymbol{\theta} = \infty$, is allowed when $\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})d\boldsymbol{\theta} < \infty$, since the corresponding posteriors are proper. However, this is an issue for the model selection with Z. Indeed, the prior $g(\boldsymbol{\theta}) = ch(\boldsymbol{\theta})$ is not completely specified, since $c > 0$ is arbitrary. Some possible solutions are given in Section 8.7.2.

Generally, the use of more diffuse (proper) priors provides smaller values of Z. Therefore, different choices of the priors can yield different selected models. For this fact, some authors criticize the use of evidence Z for model comparison.

Safe scenarios for fair comparisons

In a Bayesian framework, the best scenario is clearly when the practitioners and/or researchers have strong beliefs that can be translated into informative priors. Hence, in this setting, the priors truly encode some relevant information about the inference problem. When this additional information is not available, different strategies could be considered. We consider as a safe scenario for comparing different models, a scenario where the choice of the priors is *virtually* not favoring any of the models. Below and in Sections 8.7.2 and 8.7.4, we describe some interesting scenarios and some possible solutions for reducing, in some way, the dependence of the model comparison on the choice of the priors.

Same priors. Generally, we are interested in comparing two or more models. The use of the same (even improper) priors is possible when the models have the same parameters (and hence also share the same support space). With this choice, the resulting comparison seems fair and reasonable. However, this scenario is very restricted in practice. An example is when we have nested models. As noted in [47, Sect. 5.3], in the context of testing hypothesis, some authors have considered improper priors on nuisance parameters that appear on both null and alternative hypothesis. Since the nuisance parameters appear on both models, the multiplicative constants cancel out in the Bayes factor.

Likelihood-based priors. When $\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})d\boldsymbol{\theta} < \infty$, we can build a prior based on the data and the observation model. For instance, we can choose $g_{\text{like}}(\boldsymbol{\theta}) = \frac{\ell(\mathbf{y}|\boldsymbol{\theta})}{\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})d\boldsymbol{\theta}}$, then the marginal likelihood is

$$Z = \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta}) g_{\text{like}}(\boldsymbol{\theta}) d\boldsymbol{\theta} = \frac{\int_\Theta \ell^2(\mathbf{y}|\boldsymbol{\theta})d\boldsymbol{\theta}}{\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})d\boldsymbol{\theta}}. \tag{8.181}$$

This idea is connected to *posterior predictive approach*, described in Section 8.7.4. Indeed, the marginal likelihood above can be written as $Z = E_{P(\boldsymbol{\theta}|\mathbf{y})}[\ell(\mathbf{y}|\boldsymbol{\theta})] = \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}$ when $g(\boldsymbol{\theta}) = 1$. Less informative likelihood-based priors can be constructed using a tempering effect with a parameter $0 < \beta \leq 1$ or considering only a subset of data $\mathbf{y}_{\text{sub}}$. For instance, when $\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})^\beta d\boldsymbol{\theta} < \infty$ or $\int_\Theta \ell(\mathbf{y}_{\text{sub}}|\boldsymbol{\theta})d\boldsymbol{\theta} < \infty$, then we can choose $g_{\text{like}}(\boldsymbol{\theta}) \propto \ell(\mathbf{y}|\boldsymbol{\theta})^\beta$ or $g_{\text{like}}(\boldsymbol{\theta}) \propto \ell(\mathbf{y}_{\text{sub}}|\boldsymbol{\theta})$, the marginal likelihood is

$$Z = \frac{\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})^{\beta+1} d\boldsymbol{\theta}}{\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})^\beta d\boldsymbol{\theta}}, \quad \text{or} \quad Z = \frac{\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})\ell(\mathbf{y}_{\text{sub}}|\boldsymbol{\theta})d\boldsymbol{\theta}}{\int_\Theta \ell(\mathbf{y}_{\text{sub}}|\boldsymbol{\theta})d\boldsymbol{\theta}}. \tag{8.182}$$

This is also the key idea underlying the partial and intrinsic Bayes factors described in the next section.

8.7.2. Bayes factors with improper priors

So far we have considered proper priors, i.e., $\int_\Theta g(\boldsymbol{\theta})d\boldsymbol{\theta} = 1$. The use of improper priors is common in Bayesian inference to represent weak prior information. Consider $g(\boldsymbol{\theta}) \propto h(\boldsymbol{\theta})$ where $h(\boldsymbol{\theta})$ is a non-negative function whose integral over the state space does not converge, $\int_\Theta g(\boldsymbol{\theta})d\boldsymbol{\theta} = \int_\Theta h(\boldsymbol{\theta})d\boldsymbol{\theta} = \infty$. In that case, $g(\boldsymbol{\theta})$ is not completely specified. Indeed, we can have different definitions $g(\boldsymbol{\theta}) = ch(\boldsymbol{\theta})$ where $c > 0$ is (the inverse of) the "normalizing" constant, not uniquely determinate since c formally does not exist. Regarding the parameter inference and posterior definition, the use of improper priors poses no problems as long as $\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})h(\boldsymbol{\theta})d\boldsymbol{\theta} < \infty$, indeed

$$\begin{aligned} P(\boldsymbol{\theta}|\mathbf{y}) = \frac{1}{Z}\pi(\boldsymbol{\theta}|\mathbf{y}) &= \frac{\ell(\mathbf{y}|\boldsymbol{\theta})ch(\boldsymbol{\theta})}{\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})ch(\boldsymbol{\theta})d\boldsymbol{\theta}} = \frac{\ell(\mathbf{y}|\boldsymbol{\theta})h(\boldsymbol{\theta})}{\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})h(\boldsymbol{\theta})d\boldsymbol{\theta}}, \\ &= \frac{1}{Z_h}\ell(\mathbf{y}|\boldsymbol{\theta})h(\boldsymbol{\theta}) \end{aligned} \tag{8.183}$$

where $Z = \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})d\boldsymbol{\theta}$, $Z_h = \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})h(\boldsymbol{\theta})d\boldsymbol{\theta}$ and $Z = cZ_h$. Note that the unspecified constant $c > 0$ is canceled out, so that the posterior $P(\boldsymbol{\theta}|\mathbf{y})$ is well-defined even with an improper prior if $\int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta})h(\boldsymbol{\theta})d\boldsymbol{\theta} < \infty$. However, the issue is not solved when we compare different models, since $Z = cZ_h$ depends on c. For instance, the Bayes factors depend on the undetermined constants $c_1, c_2 > 0$ [96],

$$\mathrm{BF}(\mathbf{y}) = \frac{c_1}{c_2}\frac{\int_{\Theta_1} \ell_1(\mathbf{y}|\boldsymbol{\theta}_1)h_1(\boldsymbol{\theta}_1)d\boldsymbol{\theta}_1}{\int_{\Theta_2} \ell_2(\mathbf{y}|\boldsymbol{\theta}_2)h_2(\boldsymbol{\theta}_2)d\boldsymbol{\theta}_2} = \frac{Z_1}{Z_2} = \frac{c_1 Z_{h_1}}{c_2 Z_{h_2}}, \tag{8.184}$$

so that different choices of c_1, c_2 provide different preferable models. There exists various approaches for dealing with this issue. Below we describe some relevant ones.

Partial Bayes Factors. The idea behind the partial Bayes factors consists of using a subset of data to build proper priors and, jointly with the remaining data, they are used to calculate the Bayes factors. This is related to the likelihood-based prior approach, described above. The method starts by dividing the data in two subsets, $\mathbf{y} = (\mathbf{y}_{\text{train}}, \mathbf{y}_{\text{test}})$. The first subset $\mathbf{y}_{\text{train}}$ is used to obtain partial posterior distributions,

$$\bar{g}_m(\boldsymbol{\theta}_m|\mathbf{y}_{\text{train}}) = \frac{c_m}{Z^{(m)}_{\text{train}}}\ell_m(\mathbf{y}_{\text{train}}|\boldsymbol{\theta}_m)h_m(\boldsymbol{\theta}_m), \tag{8.185}$$

using the improper priors. The partial posterior $\bar{g}_m(\boldsymbol{\theta}_m|\mathbf{y}_{\text{train}})$ is then employed as prior. Note that

$$Z^{(m)}_{\text{train}} = c_m \int_{\Theta_m} \ell_m(\mathbf{y}_{\text{train}}|\boldsymbol{\theta}_m)h_m(\boldsymbol{\theta}_m)d\boldsymbol{\theta}_m.$$

Recall that the complete posterior of m-th model is

$$P_m(\boldsymbol{\theta}|\mathbf{y}) = P_m(\boldsymbol{\theta}|\mathbf{y}_{\text{test}}, \mathbf{y}_{\text{train}}) = \frac{c_m}{Z_m}\ell_m(\mathbf{y}|\boldsymbol{\theta}_m)h_m(\boldsymbol{\theta}_m), \tag{8.186}$$

where

$$Z_m = c_m \int_{\Theta_m} \ell_m(\mathbf{y}|\boldsymbol{\theta}_m)h_m(\boldsymbol{\theta}_m)d\boldsymbol{\theta}_m.$$

Note that $Z^{(m)}_{\text{train}}$ and Z_m both depend on the unspecified constant c_m. Considering the conditional likelihood $\ell_m(\mathbf{y}_{\text{test}}|\boldsymbol{\theta}_m, \mathbf{y}_{\text{train}})$ of the remaining data $\mathbf{y}_{\text{test}}$,[17] we can study another posterior of $\mathbf{y}_{\text{test}}$,

$$P^{(m)}_{\text{test}}(\boldsymbol{\theta}|\mathbf{y}_{\text{test}}) = \frac{1}{Z^{(m)}_{\text{test|train}}} \ell_m(\mathbf{y}_{\text{test}}|\boldsymbol{\theta}_m, \mathbf{y}_{\text{train}}) \bar{g}_m(\boldsymbol{\theta}_m|\mathbf{y}_{\text{train}}), \tag{8.187}$$

where $\bar{g}_m(\boldsymbol{\theta}_m|\mathbf{y}_{\text{train}})$ in (8.185) plays the role of a prior pdf, and

$$\begin{aligned}
Z^{(m)}_{\text{test|train}} &= \int_{\Theta_m} \ell_m(\mathbf{y}_{\text{test}}|\boldsymbol{\theta}_m, \mathbf{y}_{\text{train}}) \bar{g}_m(\boldsymbol{\theta}_m|\mathbf{y}_{\text{train}}) d\boldsymbol{\theta}_m, \\
&= \int_{\Theta_m} \ell_m(\mathbf{y}_{\text{test}}|\boldsymbol{\theta}_m, \mathbf{y}_{\text{train}}) \frac{c_m}{Z^{(m)}_{\text{train}}} \ell_m(\mathbf{y}_{\text{train}}|\boldsymbol{\theta}_m) h_m(\boldsymbol{\theta}_m) d\boldsymbol{\theta}_m, \\
&= \frac{c_m}{Z^{(m)}_{\text{train}}} \int_{\Theta_m} \ell_m(\mathbf{y}_{\text{test}}|\boldsymbol{\theta}_m, \mathbf{y}_{\text{train}}) \ell_m(\mathbf{y}_{\text{train}}|\boldsymbol{\theta}_m) h_m(\boldsymbol{\theta}_m) d\boldsymbol{\theta}_m, \\
&= \frac{c_m}{Z^{(m)}_{\text{train}}} \int_{\Theta_m} \ell_m(\mathbf{y}|\boldsymbol{\theta}_m) h_m(\boldsymbol{\theta}_m) d\boldsymbol{\theta}_m, \\
&= \frac{Z_m}{Z^{(m)}_{\text{train}}}.
\end{aligned}$$

Thus, $Z^{(m)}_{\text{test|train}}$ does not depend on c_m. Therefore, considering the partial posteriors $\bar{g}_m(\boldsymbol{\theta}_m|\mathbf{y}_{\text{train}})$ as proper priors, we can define the following *partial* Bayes factor

$$\begin{aligned}
\text{BF}(\mathbf{y}_{\text{test}}|\mathbf{y}_{\text{train}}) &= \frac{Z^{(1)}_{\text{test|train}}}{Z^{(2)}_{\text{test|train}}} = \frac{\frac{Z_1}{Z^{(1)}_{\text{train}}}}{\frac{Z_2}{Z^{(2)}_{\text{train}}}}, \\
&= \frac{\frac{Z_1}{Z_2}}{\frac{Z^{(1)}_{\text{train}}}{Z^{(2)}_{\text{train}}}} = \frac{\text{BF}(\mathbf{y})}{\text{BF}(\mathbf{y}_{\text{train}})}. \quad \textit{(“Bayes law for Bayes Factors”).}
\end{aligned} \tag{8.188}$$

Therefore, one can approximate firstly $\text{BF}(\mathbf{y}_{\text{train}})$, secondly $\text{BF}(\mathbf{y})$ and then compare the model using the partial Bayes factor $\text{BF}(\mathbf{y}_{\text{test}}|\mathbf{y}_{\text{train}})$.

Remark 22. *The trick here consists in computing two normalizing constants for each model, instead of only one. The first normalizing constant is used for building an auxiliary proper prior, depending on $\mathbf{y}_{train}$. The difference with the likelihood-based prior approach in previous section is that $\mathbf{y}_{train}$ is used only once (in the auxiliary proper prior).*

A training dataset $\mathbf{y}_{\text{train}}$ is proper if $\int_{\Theta_m} \ell_m(\mathbf{y}_{\text{train}}|\boldsymbol{\theta}_m) h_m(\boldsymbol{\theta}_i) d\boldsymbol{\theta}_m < \infty$ for all models, and it is called *minimal* if is proper and no subset of $\mathbf{y}_{\text{train}}$ is proper. If we use actually proper prior densities, the minimal training dataset is the empty set and the fractional Bayes factor reduces to the classical Bayes factor. However, the main drawback of the partial Bayes factor approach is the dependence on the choice of $\mathbf{y}_{\text{train}}$ (which could affect the selection of the model). The authors suggest finding the *minimal* suitable training set $\mathbf{y}_{\text{train}}$, but

[17] In case of conditional independence of the data given $\boldsymbol{\theta}$, we have $\ell_m(\mathbf{y}_{\text{test}}|\boldsymbol{\theta}_m, \mathbf{y}_{\text{train}}) = \ell_m(\mathbf{y}_{\text{test}}|\boldsymbol{\theta}_m)$.

this task is not straightforward. Two alternatives in the literature have been proposed, the fractional Bayes factors and the intrinsic Bayes factors.

Fractional Bayes Factors [81]. Instead of using a training data, it is possible to use power posteriors, i.e.,

$$\text{FBF}(\mathbf{y}) = \frac{\text{BF}(\mathbf{y})}{\text{BF}(\mathbf{y}|\beta)}, \tag{8.189}$$

where the denominator is

$$\text{BF}(\mathbf{y}|\beta) = \frac{\int_{\Theta_1} \ell_1(\mathbf{y}|\boldsymbol{\theta}_1)^\beta g_1(\boldsymbol{\theta}_1) d\boldsymbol{\theta}_1}{\int_{\Theta_2} \ell_2(\mathbf{y}|\boldsymbol{\theta}_2)^\beta g_2(\boldsymbol{\theta}_2) d\boldsymbol{\theta}_2} = \frac{c_1 \int_{\Theta_1} \ell_1(\mathbf{y}|\boldsymbol{\theta}_1)^\beta h_1(\boldsymbol{\theta}_1) d\boldsymbol{\theta}_1}{c_2 \int_{\Theta_2} \ell_2(\mathbf{y}|\boldsymbol{\theta}_2)^\beta h_2(\boldsymbol{\theta}_2) d\boldsymbol{\theta}_2}. \tag{8.190}$$

with $0 < \beta < 1$, and $\text{BF}(\mathbf{y}|1) = \text{BF}(\mathbf{y})$. Note that the value $\beta = 0$ is not admissible since $\int_{\Theta_m} h_m(\boldsymbol{\theta}_m) d\boldsymbol{\theta}_m = \infty$ for $m = 1, 2$. Again, since both $\text{BF}(\mathbf{y})$ and $\text{BF}(\mathbf{y}|\beta)$ depend on the ratio $\frac{c_1}{c_2}$, the fractional Bayes factor $\text{FBF}(\mathbf{y})$ is independent on c_1 and c_2 by definition.

Intrinsic Bayes factors [4]. The partial Bayes factor (8.188) will depend on the choice of (minimal) training set $\mathbf{y}_{\text{train}}$. These authors solve the problem of choosing the training sample by averaging the partial Bayes factor over all possible minimal training sets. They suggest using the arithmetic mean, leading to the *arithmetic* intrinsic Bayes factor, or the geometric mean, leading to the *geometric* intrinsic Bayes factor.

8.7.3. Marginal likelihood as a prior predictive approach

Due to the definition of the marginal likelihood $Z = E_g[\ell(\mathbf{y}|\boldsymbol{\theta})] = \int_\Theta \ell(\mathbf{y}|\boldsymbol{\theta}) g(\boldsymbol{\theta}) d\boldsymbol{\theta}$ is also called or related to the so-called *prior predictive approach*. As in the Approximate Bayesian Computation (ABC) [57], the idea is that we can generate artificial data $\widetilde{\mathbf{y}}_{i,m}$, $i = 1, ..., L$ from each m-th model with the following procedure: (a) draw $\boldsymbol{\theta}_{i,m}$ from the m-th prior, $g_m(\boldsymbol{\theta})$ and $\widetilde{\mathbf{y}}_{i,m}$ from the m-th likelihood $\ell_m(\mathbf{y}|\boldsymbol{\theta}_{i,m})$. Given each set of fake data $\mathcal{S}_m = \{\widetilde{\mathbf{y}}_{i,m}\}_{i=1}^L$, we can use different classical hypothesis testing techniques for finding the set $\mathcal{S}_m$ closest to the true data $\mathbf{y}$ (for instance, based on p-values). Another possibility, we could approximate the value $Z_m = p_m(\mathbf{y})$ applying kernel density estimation $\widehat{p}_m$ to each set $\mathcal{S}_m$.

In the next section, we describe the posterior predictive approach, which consider the expected value of likelihood evaluated in a generic $\widetilde{\mathbf{y}}$ with respect to (w.r.t.) the posterior $P(\boldsymbol{\theta}|\mathbf{y})$, instead of w.r.t. the prior $g(\boldsymbol{\theta})$. The posterior predictive idea can be considered an alternative model selection approach w.r.t. the marginal likelihood approach, which includes several well-known model selection schemes.

8.7.4. Other ways of model selection: the posterior predictive approach

The marginal likelihood approach is not the unique approach for model selection in Bayesian statistics. Here, we discuss some alternatives which are based on the concept of

prediction.
After fitting a Bayesian model, a popular approach for model checking (i.e. assessing the adequacy of the model fit to the data) consists in measuring its predictive accuracy [34, Chapter 6][83]. Hence, a key quantity in these approaches is the posterior predictive distribution of generic different data $\widetilde{\mathbf{y}}$ given $\mathbf{y}$,

$$p(\widetilde{\mathbf{y}}|\mathbf{y}) = E_{P(\boldsymbol{\theta}|\mathbf{y})}[\ell(\widetilde{\mathbf{y}}|\boldsymbol{\theta})] = \int_{\Theta} \ell(\widetilde{\mathbf{y}}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}. \tag{8.191}$$

Considering $\widetilde{\mathbf{y}} = \mathbf{y}$, note that exists a clear connection with likelihood-based priors described in Section 8.7.1.

Remark 23. *The posterior predictive distribution in* (8.191) *is an expectation w.r.t. the posterior, which is robust to the prior selection with informative data, unlike the marginal likelihood. Therefore, this approach is less affected by the prior choice.*

Note that we can consider posterior predictive distributions $p(\widetilde{\mathbf{y}}|\mathbf{y})$ for vectors $\widetilde{\mathbf{y}}$ smaller than $\mathbf{y}$ (i.e., with less components). The *posterior predictive checking* is based on the main idea of considering some simulated data $\widetilde{\mathbf{y}}_i \sim p(\widetilde{\mathbf{y}}|\mathbf{y})$, with $i = 1, ..., L$, and comparing with the observed data $\mathbf{y}$. After obtaining a set of fake data $\{\widetilde{\mathbf{y}}_i\}_{i=1}^L$, we have to measure the discrepancy between the true observed data $\mathbf{y}$ and the set $\{\widetilde{\mathbf{y}}_i\}_{i=1}^L$. This comparison can be made with test quantities and graphical checks (e.g., posterior predictive p-values).

Alternatively, different measures of predictive accuracy can be employed. An example, is the *expected log pointwise predictive density* (ELPD) [102]. Let recall that $\mathbf{y} = [y_1, \ldots, y_{D_y}] \in \mathbb{R}^{D_y}$, and define as $\bar{y} \in \mathbb{R}$ any alternative scalar data. Considering M alternative scalar data $\bar{y}_i$ with density $p_{\text{true}}(\bar{y}_i)$, the ELPD is defined as

$$\begin{aligned} \text{ELPD} &= \sum_{i=1}^{M} \int_{\mathbb{R}} \log p(\bar{y}_i|\mathbf{y}) p_{\text{true}}(\bar{y}_i) d\bar{y}_i \\ &= \sum_{i=1}^{M} \int_{\mathbb{R}} \log \left[\int_{\Theta} \ell(\bar{y}_i|\boldsymbol{\theta}) P(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \right] p_{\text{true}}(\bar{y}_i) d\bar{y}_i. \end{aligned} \tag{8.192}$$

Note that $p_{\text{true}}(\bar{y}_i)$ is the density representing the true data generating process for $\bar{y}_i$, which is clearly unknown. Therefore, some approximations are required. First all, we define an over-estimation of the ELPD, considering the observed data in $\mathbf{y} = [y_1, \ldots, y_{D_y}]$ instead new alternative data $\bar{y}_i$, so that $M = D_y$ and $\int_{\mathbb{R}} \log p(\bar{y}_i|\mathbf{y}) p_{\text{true}}(\bar{y}_i) d\bar{y}_i \approx \log p(y_i|\mathbf{y})$, i.e.,

$$\widehat{\text{ELPD}} = \sum_{i=1}^{D_y} \log p(y_i|\mathbf{y}) = \sum_{i=1}^{D_y} \log \left[\int_{\Theta} \ell(y_i|\boldsymbol{\theta}) P(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \right]. \tag{8.193}$$

In practice, we need an additional approximation for computing $p(y_i|\mathbf{y}) = \int_{\Theta} \ell(\bar{y}_i|\boldsymbol{\theta}) P(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta}$. We can use MCMC samples from $P(\boldsymbol{\theta}|\mathbf{y})$, i.e.,

$$\widehat{\text{ELPD}} = \sum_{i=1}^{D_y} \log \widehat{p}(y_i|\mathbf{y}) = \sum_{i=1}^{D_y} \log \left[\frac{1}{N} \sum_{n=1}^{N} \ell(y_i|\boldsymbol{\theta}_n) \right], \quad \text{with} \quad \boldsymbol{\theta}_n \sim P(\boldsymbol{\theta}|\mathbf{y}). \tag{8.194}$$

LOO-CV. However, we know that the approximation above overestimates ELPD. One possibility is to use cross-validation (CV), such as the leave-one-out cross-validation (LOO-CV). In LOO-CV, we consider $p(y_i|\mathbf{y}_{-i})$ instead of $p(y_i|\mathbf{y})$ in Eq. (8.193), where $\mathbf{y}_{-i}$ is vector $\mathbf{y}$ leaving out the i-th data, y_i. Hence,

$$\widehat{\text{ELPD}}_{\text{LOO-CV}} = \sum_{i=1}^{D_y} \log p(y_i|\mathbf{y}_{-i}) = \sum_{i=1}^{D_y} \log \left[\int_\Theta \ell(y_i|\boldsymbol{\theta}) P(\boldsymbol{\theta}|\mathbf{y}_{-i}) d\boldsymbol{\theta} \right]. \tag{8.195}$$

For approximating $p(y_i|\mathbf{y}_{-i}) = \int_\Theta \ell(y_i|\boldsymbol{\theta}) P(\boldsymbol{\theta}|\mathbf{y}_{-i}) d\boldsymbol{\theta}$, we draw again from the full posterior by means of an MCMC technique, $\boldsymbol{\theta}_n \sim P(\boldsymbol{\theta}|\mathbf{y})$, and apply importance sampling [102],

$$p(y_i|\mathbf{y}_{-i}) \approx \widehat{p}(y_i|\mathbf{y}_{-i}) = \sum_{n=1}^{N} \bar{w}_{i,n} \ell(y_i|\boldsymbol{\theta}_n), \qquad \boldsymbol{\theta}_n \sim P(\boldsymbol{\theta}|\mathbf{y}), \tag{8.196}$$

where $\bar{w}_{i,n} = \frac{w_{i,n}}{\sum_{k=1}^{N} w_{i,k}}$ and, in the case the data are conditionally independent,

$$w_{i,n} = \frac{1}{\ell(y_i|\boldsymbol{\theta}_n)} \propto \frac{P(\boldsymbol{\theta}_n|\mathbf{y}_{-i})}{P(\boldsymbol{\theta}_n|\mathbf{y})}.$$

Thus, replacing in (8.196), we obtain

$$p(y_i|\mathbf{y}_{-i}) \approx \widehat{p}(y_i|\mathbf{y}_{-i}) = \frac{1}{\sum_{n=1}^{N} \frac{1}{\ell(y_i|\boldsymbol{\theta}_n)}}, \qquad \boldsymbol{\theta}_n \sim P(\boldsymbol{\theta}|\mathbf{y}), \tag{8.197}$$

which resembles the harmonic mean estimator but with just one data point. However, since the full posterior $P(\boldsymbol{\theta}_n|\mathbf{y})$ has smaller variance of $P(\boldsymbol{\theta}_n|\mathbf{y}_{-i})$, the direct use of (8.197) is quite unstable, since the IS weights can have high or infinite variance. See [102] for stable computations of LOO-CV and using posterior simulations. Moreover, see also [83] for a quantitative comparison of methods for estimating the predictive ability of a model. The marginal likelihood can also be interpreted as a measure of predictive performance [47, Sect. 3.2]. In [29], the authors show that the marginal likelihood is equivalent, in some sense, to a leave-p-out cross-validation procedure. For further discussions about model selection strategies, see [1, 82].

8.8. Numerical comparisons

In this section, we compare the performance of different marginal likelihood estimators in different experiments. First of all, we consider 3 different illustrative scenarios in Section 8.8.1, 8.8.2 and 8.8.3 each one considering different challenges: different overlap between prior and likelihood (changing the number of data, or the variance and mean of the prior), multi-modality and different dimensions of the inference problem. The first experiment also considers two different sub-scenarios. Additional theoretical results related to the experiments in Sect. 8.8.1 are provided in the Supplementary Material.

The last two experiments involves a real data analysis. In Section 8.8.4, we test several estimators in a nonlinear regression problem with real data (studied also in [23]), where

the likelihood function has non-elliptical contours. Finally, in Section 8.8.5 we consider another regression problem employing non-linear localized bases with real data of the COVID-19 outbreak.

8.8.1. First experiment

First setting: Gaussians with same mean and different variances

In this example, our goal is to compare by numerical simulations different schemes for estimating the normalizing constant of a Gaussian target $\pi(\theta) = \exp(-\frac{1}{2}\theta^2)$. We know the ground-truth $Z = \int_{-\infty}^{\infty} \pi(\theta)d\theta = \sqrt{2\pi}$, so $P(\theta) = \frac{\pi(\theta)}{Z} = \mathcal{N}(\theta|0,1)$. Since this is a data-independent example, $\pi(\theta)$ and $P(\theta)$ have no dependence on $\mathbf{y}$. We compare several estimators enumerated below, considering one or two proposals.

One proposal estimators (IS and RIS). First of all, we recall that the IS vers-1 estimator with importance density $\bar{q}(\theta)$ and the RIS estimator with auxiliary density $f(\theta)$ are

$$\widehat{Z}_{\text{IS}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\pi(z_i)}{\bar{q}(z_i)}, \quad z_i \sim \bar{q}(\theta), \quad \widehat{Z}_{\text{RIS}} = \frac{1}{\frac{1}{N}\sum_{i=1}^{N}\frac{f(\theta_i)}{\pi(\theta_i)}}, \quad \theta_i \sim P(\theta).$$

For a fair comparison, we consider

$$\bar{q}(\theta) = f(\theta) = \mathcal{N}(\theta|0,h^2) = \frac{1}{\sqrt{2\pi h^2}}\exp\left(-\frac{1}{2h^2}\theta^2\right).$$

where $h > 0$ is the standard deviation. We desire to study the performance of the two estimators as h varies. Moreover, a theoretical comparison of IS and RIS estimators is given in the Supplementary Material.

Estimators using with two proposals. The IS and RIS estimators use a single set of samples from $\bar{q}(\theta)$ or $P(\theta)$, respectively. Now, we consider the comparison, in terms of MSE, against several estimators that use sets of samples from both densities, $\bar{q}(\theta)$ and $P(\theta)$, at the same time. Let $\{z_i\}_{i=1}^{M}$ and $\{\theta_j\}_{j=1}^{N}$ denote sets of iid samples from $\bar{q}(\theta)$ and $P(\theta)$, respectively. When $M = N = 500$, the set $\{\{z_i\}_{i=1}^{M}, \{\theta_j\}_{j=1}^{N}\}$ can be considered as a unique set of samples drawn from the mixture $\frac{1}{2}P(\theta) + \frac{1}{2}\bar{q}(\theta)$ [28]. For a fair comparison, these estimators use $\frac{M}{2}$ samples from $\bar{q}(\theta)$ and $\frac{N}{2}$ samples from $P(\theta)$.

Ideal and realistic scenarios. Furthermore, we consider two scenarios, corresponding to whether we can evaluate $P(\theta)$ (ideal and impossible scenario) or we evaluate $\pi(\theta) \propto P(\theta)$ (realistic scenario). Note that the first scenario is simply for illustration purposes.

Jointly with IS and RIS estimator, we test several other estimators of Z, introduced in Section 8.4.2, that use two sets of samples simultaneously.

- **Opt-BS:** The optimal bridge sampling estimator with $\alpha(\theta) = (\frac{1}{2}P(\theta) + \frac{1}{2}\bar{q}(\theta))^{-1}$.

- **Mix-IS:** IS vers-1 with the mixture $\frac{1}{2}P(\theta) + \frac{1}{2}\bar{q}(\theta)$, instead of $\bar{q}(\theta)$, as proposal.
- **Mix-self IS:** The self-IS estimator, with $f(\theta) = \bar{q}(\theta)$, and the mixture $\frac{1}{2}P(\theta) + \frac{1}{2}\bar{q}(\theta)$ as the proposal.

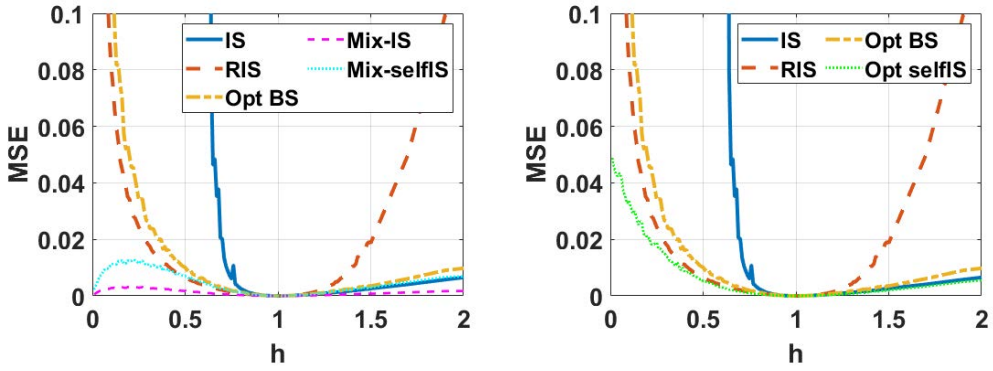
Moreover, we consider another one proposal estimator, described in Section 8.4.1:

- **Opt-self IS:** The optimal self-IS estimator, with $f(\theta) = \bar{q}(\theta)$. Note that this estimator use samples from a density to $\bar{q}^{\text{opt}}(\theta) \propto |P(\theta) - \bar{q}(\theta)|$. We include it as a reference, for its optimality, and since $\bar{q}^{\text{opt}}(\theta)$ involves both, $P(\theta)$ and $\bar{q}(\theta)$.

Remark 24. *Clearly, in the realistic scenario, all of the schemes above must replaced for their iterative versions, since we cannot evaluate $P(\theta)$ but only $\pi(\theta) \propto P(\theta)$.*

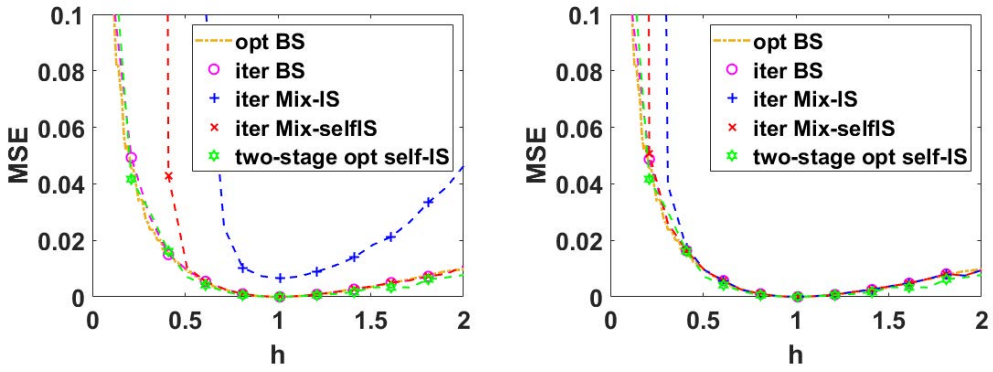
Results in ideal scenario. Figures 8.8(a)-(b) show the MSE of the estimators versus h (which is the standard deviation of $\bar{q}(\theta)$) in the ideal scenario. IS vers-1 can have very high MSE when $h < 1$, i.e., $\bar{q}(\theta)$ has smaller variance then the $P(\theta)$. Whereas, IS vers-1 is quite robust when $h > 1$. The MSE of RIS has the opposite behavior of IS vers-1. This is because RIS needs that $\bar{q}(\theta)$ has lighter tails than $P(\theta)$. In this example, optimal bridge sampling seems to provide performance in-between the IS and RIS estimators. The MSE of Opt-BS is closer to RIS for $h < 1$, whereas Opt-BS becomes closer to IS for $h > 1$. Conversely, the MSE of Opt BS is not smaller than that of IS or RIS for any h in this example. Finally, Mix-IS and Mix-self-IS provide the best performance, even better than the optimal self-IS estimator. But this is due to we are in an ideal, unrealistic scenario.

Results in the realistic scenario. Since Z is unknown we cannot evaluate $P(\theta)$ but only $\pi(\theta) \propto P(\theta)$. Only IS and RIS can be truly applied. The rest of above estimators must employ an iterative procedure (see Section 8.4.1 and Section 8.4.2). The iterative versions of these estimators evaluate $\frac{1}{2}\pi(\theta)/\widehat{Z}^{(t)} + \frac{1}{2}\bar{q}(\theta)$, where $\widehat{Z}^{(t)}$ is the current approximation. In Figure 8.9, we show these three estimators after $T = 5$ and $T = 15$ iterations. Interestingly, note that they all converge to the results of Opt-BS estimator. This means that the iterative versions of Mix-IS and Mix-self-IS are two alternative of Opt-BS in practice, and the performance obtained in ideal scenario are unachievable. However, the iterative version of Opt-BS seems to have the fastest convergence (to the results of the ideal Opt-BS), w.r.t. the iterative versions of Mix-IS and Mix-self-IS.
We also include a two-stage version of the Opt-selfIS estimator (see Section 8.4.1). This estimator employs $\frac{N}{4}$ to obtain an approximation $\widehat{Z}$ via standard IS, and then draws $\frac{3}{4}N$ samples from a density proportional to $|\pi(\theta)/\widehat{Z} - \bar{q}(\theta)|$. This two-stage Opt-selfIS depends on the quality of the initial approximation of $\widehat{Z}$. Since this initial approximation is provided by IS, and since IS is problematic when $h < 1$, the two-stage self-IS does not perform better than Opt-BS for $h < 1$.



(a) (b)

Figure 8.8: Numerical comparison with estimators using samples from $\bar{q}(\theta)$ and $P(\theta)$, and optimal self-IS. The figure shows the MSE of each method (averaged over 2000 simulations) as a function of h.



(a) T=5 (b) T=15

Figure 8.9: Comparison of iterative version of estimators with a very far starting value, $\widehat{Z}^{(0)} = 5000$, with $T = 5$ and $T = 15$. Note that the two-stage self-IS is not iterative (see Section 8.4.1).

Second setting: Gaussians with same variance and different means

In this setting, we consider again $P(\theta) \propto \pi(\theta) = \exp\left(-\frac{1}{2}\theta^2\right)$,i.e., $P(\theta) = \frac{\pi(\theta)}{Z} = \mathcal{N}(\theta|0,1)$, but the proposal is $\bar{q}(\theta) = \mathcal{N}(\theta|\mu,1)$ for $\mu \geq 0$. Namely, as μ grows, $\bar{q}(\theta)$ and $P(\theta)$ are more distant. A theoretical comparison of IS and RIS estimators is given in the Supplementary Material, also for this setting.

Similarly, we compare the MSE as function of μ of different estimators of Z: (a) IS vers-1, (b) RIS, (c) optimal BS (Opt-BS), (d) a suboptimal self-IS estimator with $f(\theta) = \bar{q}(\theta)$ and using $\bar{q}(\theta) = \mathcal{N}(\frac{\mu}{2},1)$ as proposal, and (e) the Opt-self IS estimator with $f(\theta) = \bar{q}(\theta)$ and proposal $\bar{q}^{\text{opt}}(\theta) \propto |P(\theta) - \bar{q}(\theta)|$. Each estimator is computed using 500 samples in total and the results are averaged over 2000 independent simulations.

Results of the second setting. Unlike in the first setting, here we consider only the ideal scenario (i.e., without iterative procedures). However, note that the suboptimal self-IS scheme would not require an iterative version. The results are shown in Figure 8.10. The MSE of both IS and RIS diverge as e^{μ^2}. Opt-BS shows better performance than IS vers-1 and RIS. The suboptimal self-IS estimator performs similarly to the Opt-BS, but both are worse than the Opt-self IS estimator. In this example, the estimators that use a middle density (as Opt-BS and the self-IS estimators) are less affected by the problem of $P(\theta)$ and $\bar{q}(\theta)$ becoming further apart. As in the previous setting, we expect that the iterative versions of Opt-BS converges to the results of the ideal Opt-BS, provided in Figure 8.10. Recall that, for approximating the Opt-self IS, we require a two-stage procedure. However, a procedure with just two stages could be not enough, as we showed in the previous setting. Hence, an iterative application of the two-stage procedure could be employed (becoming actually an adaptive importance sampler).

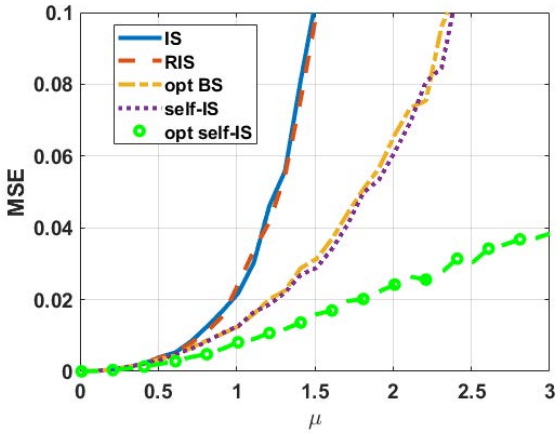


Figure 8.10: Numerical comparison of IS, RIS, Opt-BS, suboptimal self-IS and Opt self-IS. The figure shows the MSE of each method (averaged over 2000 simulations) as a function of μ. Greater μ means $P(\theta)$ and $\bar{q}(\theta)$ are further apart.

8.8.2. Second experiment: Gaussian likelihood and uniform prior

Let us consider the following one-dimensional example. More specifically, we consider independent data $\mathbf{y} = [y_1, \dots, y_{D_y}]$ generated according to a Gaussian observation model,

$$\ell(\mathbf{y}|\theta) = \prod_{i=1}^{D_y} \ell(y_i|\theta) = \frac{1}{(\sqrt{2\pi}\sigma)^{D_y}} \exp\left\{-\frac{D_y}{2\sigma^2}[(\theta - \bar{y}) + s_y]\right\},$$

where $\sigma = 3$, $\bar{y}$ and s_y denote the sample mean and sample variance of $\mathbf{y}$, respectively. We consider a uniform prior $g(\theta) = \frac{1}{2\Delta}$, $\theta \in [-\Delta, \Delta]$ with $\Delta > 0$ being the prior width. In this setting, the marginal likelihood Z can be obtained in closed-form as a function of Δ and n (considering the evaluation of the error function $\text{erf}(x)$). The posterior is a truncated

Gaussian $P(\theta|\mathbf{y}) \propto \mathcal{N}(\theta|\bar{y}, \frac{\sigma^2}{D_y})$, $\theta \in [-\Delta, \Delta]$. Let $\beta \in [0, 1]$ denote an inverse temperature, the power posterior is

$$P(\theta|\mathbf{y}, \beta) \propto \mathcal{N}\left(\theta\middle|\bar{y}, \frac{\sigma^2}{D_y\beta}\right), \quad \text{restricted to } \theta \in [-\Delta, \Delta]. \tag{8.198}$$

For any β, we can sample $P(\theta|\mathbf{y}, \beta) \propto \ell(\mathbf{y}|\theta)^\beta g(\theta)$ with rejection sampling by drawing from $\mathcal{N}(\theta|\bar{y}, \frac{\sigma^2}{D_y\beta})$ and discarding the samples that fall outside $[-\Delta, \Delta]$.

Scenario 1: $\Delta = 10$ and $D_y = 10$. We start by setting $\Delta = 10$ and generating $D_y = 10$ data points from $\mathcal{N}(0, \sigma^2)$ with $\sigma = 3$. The value of the marginal likelihood is $\log Z = -25.2853$. We aim to compare the performances of several methods in estimating $\log Z$: (a) Naive Monte Carlo (NMC), (b) Harmonic mean (HM), (c) IS with a tempered posterior as proposal (IS-P), (d) stepping stone sampling (SS), (e) power posterior method (PP), and (f) path sampling (PS).

Remark 25. *Estimating* $\log Z$*, instead of directly* Z*, helps the methods of PP and PS, with respect to NMC, HM, IS-P and SS (making their results worse).*

We establish a total budget of $N = 10^3$ likelihood evaluations. For SS and PP, we set $K+1$ values of β, from $\beta_0 = 0$ to $\beta_K = 1$, chosen (i) uniformly, i.e., $\beta_k = \frac{k}{K}$ for $k = 1, \ldots, K$, or (ii) concentrated around $\beta = 0$, i.e., $\beta_k = \left(\frac{k}{K}\right)^{1/\alpha}$ with $\alpha = 0.25$. Hence the uniform case is obtained when $\alpha = 1$. Note that SS draws samples from K distributions, while PP draw samples from $K + 1$ distributions. For fair comparison, we sample $\lfloor \frac{N}{K} \rfloor$ times from each $P(\theta|\mathbf{y}, \beta_k)$, for $k = 0, \ldots, K-1$, in SS, and $\lfloor \frac{N}{K+1} \rfloor$ times from of each $P(\theta|\mathbf{y}, \beta_k)$, for $k = 0, \ldots, K$, in PP. For IS-P we test $\beta_1 = 0.5$ and $\beta_2 = 0.5^4$ and draw N samples from each of the $P(\theta|\mathbf{y}, \beta_1)$ and $P(\theta|\mathbf{y}, \beta_2)$. For PS, we sample N pairs (β', θ') as follows: we first sample β' from a $\mathcal{U}(0, 1)$ and then sample θ' from the corresponding power posterior $P(\theta|\mathbf{y}, \beta')$. Naive Monte Carlo uses N independent samples from prior and HM uses N independent samples from the posterior.

Results scenario 1. In Figure 8.11(a), we show 500 independent estimations from each method. We observe that NMC works very well in this scenario since the prior acts as a good proposal. SS with $K = 2$ provides also good performance, since half the samples come from the prior with this choice of K. The value of α seems to be not important for SS in this case. PS performs as well as NMC and SS, but shows a slightly bigger dispersion. HM tends to overestimate the marginal likelihood, which is a well-known issue. The estimation provided by IS-P depends on the choice of β. For $\beta_1 = 0.5$, the power posterior is closer to the posterior so its behavior is similar to HM. For $\beta_2 = 0.0625$ the power posterior is close to the prior, and IS-P tends to underestimate Z. Recall that IS-P has a bias since it is a special case of IS vers-2. PP performs poorly with $K = 2$, due to the discretization error in (8.103), which improves when considering the value $K = 35$. The choice $\alpha = 0.25$, w.r.t. $\alpha = 1$, improves the performance in PP.
In Figure 8.11(b), we show the mean absolute error (MAE) in estimating $\log Z$ of SS

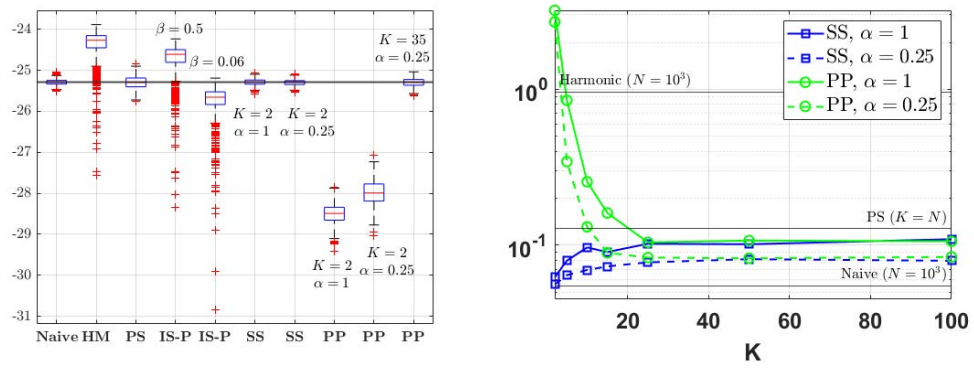
and PP as a function of K. We depict two curves for each method, corresponding to the choices $\alpha = 1$ and $\alpha = 0.25$. We can observe that the errors obtained in SS and PP when $\alpha = 0.25$ are smaller than when $\alpha = 1$ for any K. This is in line with the recommendations provided in their original works. We note that the error of SS slightly deteriorates as K grows: for $K > 2$, less and less samples are drawn from the prior, which is a good proposal in this scenario (with $\Delta = 10$ and $D_y = 10$). The performance of PP improves drastically as K grows, since larger K means that the trapezoidal rule is more accurate in approximating (8.103). SS and PP, for $\alpha = 1$ and $\alpha = 0.25$, approach the same limit when K grows, achieving an error which is always greater than the one obtained by NMC, in this scenario.

Scenario 2: $\Delta = 1000$ and $D_y = 100$. Now, we replicate the previous experiment increasing the number of data, $D_y = 100$, and the width of the prior, $\Delta = 1000$. The joint effect of increasing D_y and Δ makes the likelihood become extremely concentrated w.r.t. the prior, hence decreasing the value of the marginal likelihood, being $\log Z = -267.6471$. Moreover, this high discrepancy between prior and posterior is reflected in the power posteriors $P(\theta|\mathbf{y}, \beta)$, which will be very similar to the posterior except for very small values of β. We compare all the methods described before with a total budget of $N = 10^3$ likelihood evaluations. Additionally, we also test a PS where $\beta' \sim \mathcal{B}(0.25, 1)$, i.e., from a beta distribution which provides more β' values closer to 0.

Results scenario 2. In Figure 8.12(a), we can see that, unlike in the previous scenario, the NMC tends to underestimate the marginal likelihood, since the likelihood is much more concentrated than the prior. The HM and the two implementations of IS-P provide similar results, overestimating Z: in this case, the posterior is so different from the prior that $P(\theta|\mathbf{y}, 0.5)$ and $P(\theta|\mathbf{y}, 0.06)$ are very similar to the posterior. PS with $\beta' \sim [0, 1]$ tends to overestimate Z: since the β''s are drawn uniformly in $[0, 1]$, many samples (β', θ') are drawn in high-valued likelihood zones. Indeed, at least the bias is reduced when we test PS with $\beta' \sim \mathcal{B}(0.25, 1)$. We also show the results of one implementation of SS (with $K = 10$ and $\alpha = 0.25$) and PP (with $K = 70$ and $\alpha = 0.25$). Both greatly outperform the rest of estimators in this scenario, providing accurate estimations. In Figure 8.12(b), we show again the MAE of SS and PP as a function of K for two values $\alpha = 1$ and $\alpha = 0.25$. The error of PP, with either $\alpha = 1$ or $\alpha = 0.25$, decreases as K grows, although it decreases more rapidly when considering $\alpha = 0.25$. The error of SS with $\alpha = 0.25$ decreases as K grows, but increases with K when $\alpha = 1$. Again, PP requires the use bigger values of K with respect to SS. In both methods, the choice of $\alpha < 1$, i.e., concentrating β's near $\beta = 0$ where $P(\theta|\mathbf{y}, \beta)$ is usually changing rapidly, shows to improve the overall performance.

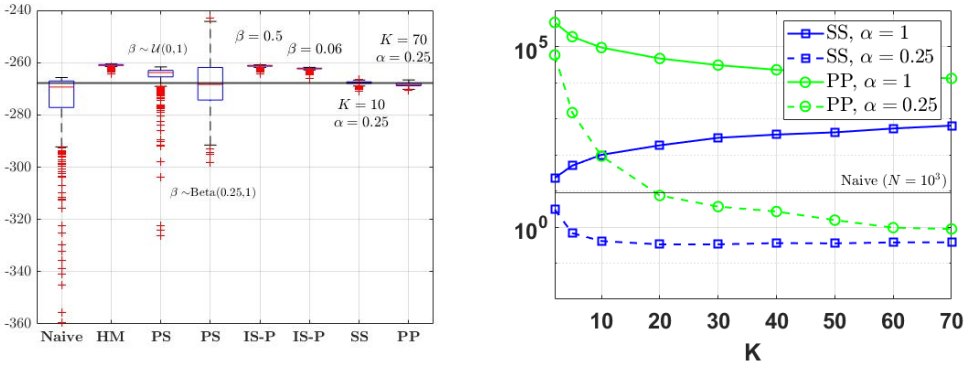
8.8.3. Third experiment: posterior as mixture of two components

We consider a posterior which is a mixture of two D_θ-dimensional Gaussian densities. It is a conjugate model where the likelihood is Gaussian and the prior is a mixture of two Gaussian. Given the observation vector $\mathbf{y}$, we consider a D_θ-dimensional Gaussian



(a) (b)

Figure 8.11: Simulations when $D_y = 10$ and $\Delta = 10$: (a) Estimates of $\log Z$ in 500 independent simulations, (b) MAEs of SS and PP as a function of K for two values of α.



(a) (b)

Figure 8.12: Simulations when $D_y = 100$ and $\Delta = 1000$: (a) Estimates of $\log Z$ in 500 independent simulations, (b) MAEs of SS and PP as a function of K for two values of α.

likelihood function

$$\ell(\mathbf{y}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\Lambda}), \tag{8.199}$$

with covariance $\boldsymbol{\Lambda}$, and a D_θ-dimensional Gaussian mixture prior

$$g(\boldsymbol{\theta}) = \alpha_{\text{prior}}\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\text{pr}}^{(1)}, \boldsymbol{\Sigma}_{\text{pr}}^{(1)}) + (1 - \alpha_{\text{prior}})\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\text{pr}}^{(2)}, \boldsymbol{\Sigma}_{\text{pr}}^{(2)}), \tag{8.200}$$

with $\alpha_{\text{prior}} \in [0, 1]$, $\boldsymbol{\mu}_{\text{pr}}^{(i)}$ and $\boldsymbol{\Sigma}_{\text{pr}}^{(i)}$ being the prior means and covariances of each component of the mixture, respectively. Then, the posterior is also a mixture of two Gaussian densities

$$P(\boldsymbol{\theta}|\mathbf{y}) = \alpha_{\text{post}}\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\text{post}}^{(1)}, \boldsymbol{\Sigma}_{\text{post}}^{(1)}) + (1 - \alpha_{\text{post}})\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\text{post}}^{(2)}, \boldsymbol{\Sigma}_{\text{post}}^{(2)}), \tag{8.201}$$

where the parameters $\alpha_{\text{post}} \in [0, 1]$, $\boldsymbol{\mu}_{\text{post}}^{(i)}$ and $\boldsymbol{\Sigma}_{\text{post}}^{(i)}$ can be obtained in closed-form from α_{prior}, $\boldsymbol{\mu}_{\text{pr}}^{(i)}$, $\boldsymbol{\Sigma}_{\text{pr}}^{(i)}$, $\boldsymbol{\Lambda}$ and $\mathbf{y}$. Thus, having the analytical expression of the posterior in closed-form allows to compute exactly the marginal likelihood Z (recall $Z = \frac{\pi(\boldsymbol{\theta}|\mathbf{y})}{P(\boldsymbol{\theta}|\mathbf{y})}$ for any $\boldsymbol{\theta}$).

In this case, we can also draw samples directly from the posterior. We can interpret this scenario as the use of an ideal MCMC scenario, where the performance is extremely good. We compare different estimators of Z changing the Euclidean distance between the means of posterior mixture components,

$$\texttt{dist} = \|\boldsymbol{\mu}_{\text{post}}^{(1)} - \boldsymbol{\mu}_{\text{post}}^{(2)}\|_2, \tag{8.202}$$

in $D_\theta = 1$ and $D_\theta = 5$. This distance can be controlled by changing the distance between the prior modes. More specifically, we choose $\boldsymbol{\Lambda} = 50\mathbf{I}_D$, $\boldsymbol{\Sigma}_{\text{pr}}^{(1)} = \boldsymbol{\Sigma}_{\text{pr}}^{(2)} = 30\mathbf{I}_D$, where $\mathbf{I}_D$ denotes the D-dimensional identity matrix. The data is a single observation $\mathbf{y} = -0.5\mathbf{1}_D$, where $\mathbf{1}_D$ a D-dimensional vector of 1's. For the prior means we chose $\boldsymbol{\mu}_{\text{pr}}^{(1)} = -\boldsymbol{\mu}_{\text{pr}}^{(2)} = L\mathbf{1}_D$, so $\left\|\boldsymbol{\mu}_{\text{pr}}^{(1)} - \boldsymbol{\mu}_{\text{pr}}^{(2)}\right\|_2 = 2L\sqrt{D_\theta}$. We can change the distance between the modes of the prior, and hence between the modes of the posterior, by varying $L \in \mathbb{R}^+$. Specifically, we select $L \in \{1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 51\}$ and compare: (i) the Naive-MC estimator, (ii) the HM estimator, (iii) Laplace-Metropolis estimator, (iv) RIS, and (v) CLAIS. The budget is 10^4 posterior evaluations. In RIS, we set $f(\boldsymbol{\theta})$ to be the mixture in Eq. (8.146), that results after applying a clustering algorithm (e.g., k-means algorithm) to the 10^4 posterior samples. In CLAIS, we use an analogous mixture obtained from $5 \cdot 10^3$ posterior samples, and then use it to draw other $5 \cdot 10^3$ samples in the lower layer (hence the total number of posterior evaluations is 10^4). For RIS and CLAIS, we set the number of clusters to $C = 4$. RIS and CLAIS also need setting the bandwidth parameter h (see Eq. (8.146)). We find that the choices $h = 2$ for RIS and $h = 10$ for CLAIS show the average performance of both. We test the techniques in dimension $D_\theta = 1$ and $D_\theta = 5$. We compute the relative Mean Absolute Error (MAE) in the estimation of Z, averaged over 200 independent simulations.

The results are depicted in Figure 8.13. They show that RIS and the CLAIS achieve the best overall performances. Their relative error remain small and rather constant for all distances considered, for $D_\theta = 1$ and $D_\theta = 5$. The RIS estimator performs as well as CLAIS in both $D_\theta = 1$ and $D_\theta = 5$, and even better for small distances in $D_\theta = 1$. For the smallest distance, the lowest relative error corresponds to the Naive MC estimator, since prior and posterior are very similar in that case, although it rapidly gets outperformed by RIS and CLAIS. The Laplace estimator provides poor results as `dist` grows, since the posterior becomes bimodal. As one could expect, the estimators that make use of the posterior sample to adapt its importance density, i.e., RIS and CLAIS, achieve best performances, being almost independent to increasing the distance between the modes. The HM estimator confirms its reputation of relative bad estimator.

8.8.4. Experiment with biochemical oxygen demand data

We consider a numerical experiment studied also in [23], that is a nonlinear regression problem modeling data on the biochemical oxygen demand (BOD) in terms of time instants. The outcome variable Y_i = BOD (mg/L) is modeled in terms of t_i = time (days)

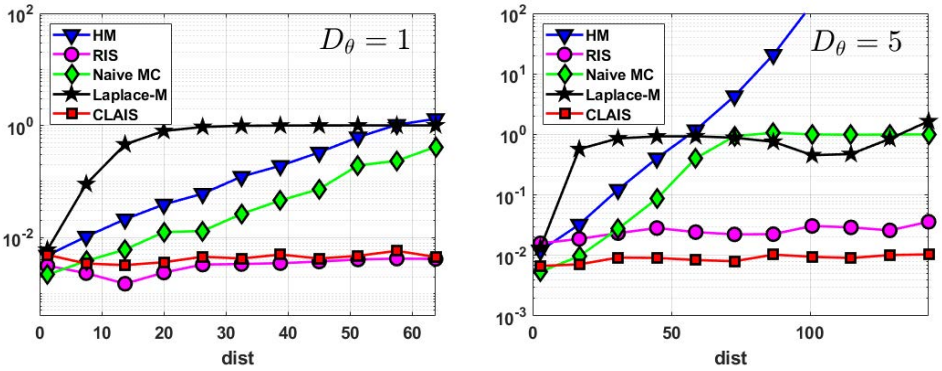


Figure 8.13: Relative MAE versus `dist` in dimension $D_\theta = 1$ and dimension $D_\theta = 5$.

as

$$Y_i = \theta_1(1 - e^{-\theta_2 t_i}) + \epsilon_i, \quad i = 1, \ldots, 6, \tag{8.203}$$

where the ϵ_i's are independent $\mathcal{N}(0, \sigma^2)$ errors, hence $Y_i \sim \mathcal{N}(\theta_1(1 - e^{-\theta_2 t_i}), \sigma^2)$. The data $\{y_i\}_{i=1}^6$, measured at locations $\{t_i\}_{i=1}^6$, are shown in Table 8.18 below.

Table 8.18: Data of the numerical experiment in Section 8.8.4.

t_i (days)	y_i (mg/L)
1	8.3
2	10.3
3	19.0
4	16.0
5	15.6
7	19.8

The goal is to compute the normalizing constant of the posterior of $\boldsymbol{\theta} = [\theta_1, \theta_2]$ given the data $\mathbf{y} = \{(t_i, y_i)\}_{i=1}^6$. Following [23], we consider uniform priors for $\theta_1 \sim \mathcal{U}([0, 60])$, and $\theta_2 \sim \mathcal{U}([0, 6])$, i.e., $g_1(\theta_1) = \frac{1}{60}$ for $\theta_1 \in [0, 60]$, and $g_2(\theta_2) = \frac{1}{6}$, with $\theta_2 \in [0, 6]$. Moreover, we consider an improper prior for σ, $g_3(\sigma) \propto \frac{1}{\sigma}$. However, we will integrate out the variable σ. Indeed, the two-dimensional target $\pi(\boldsymbol{\theta}|\mathbf{y}) = \pi(\theta_1, \theta_2|\mathbf{y})$ results after integrating out σ by marginalizing

$$\pi(\theta_1, \theta_2, \sigma|\mathbf{y}) = \ell(\mathbf{y}|\theta_1, \theta_2, \sigma) g_1(\theta_1) g_2(\theta_2) g_3(\sigma),$$

w.r.t. σ, namely we obtain

$$\pi(\boldsymbol{\theta}|\mathbf{y}) = \int \pi(\theta_1, \theta_2, \sigma|\mathbf{y}) d\sigma = \ell(\mathbf{y}|\theta_1, \theta_2)\, g_1(\theta_1) g_2(\theta_2) \tag{8.204}$$

$$= \frac{1}{60}\frac{1}{6}\frac{1}{\pi^3} \frac{8}{\left\{\sum_{i=1}^6 [y_i - \theta_1(1 - \exp(-\theta_2 t_i))]^2\right\}^3}, \quad [\theta_1, \theta_2] \in [0, 60] \times [0, 6], \tag{8.205}$$

for which we want to compute its normalizing constant $Z = \int \pi(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}$. The derivation is given in the Supplementary Material. The true value (ground-truth) is $\log Z = -16.208$, considering the data in Table 8.18.

Scenario 1. As in [23], we compare the relative MAE, $\frac{\mathbb{E}\left[|\widehat{Z}-Z|\right]}{Z}$, obtained by different methods: (a) the naive Monte Carlo estimator; **(b)** a modified version of the Laplace method (more sophisticated) given in [23]; **(c)** the Laplace-Metropolis estimator in Sect. 8.3.1 (using sample mean and sample covariance considering MCMC samples from $P(\boldsymbol{\theta}|\mathbf{y})$); **(d)** the HM estimator of Eq. 8.46; **(f)** the RIS estimator where $f(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance of the MCMC samples from $P(\boldsymbol{\theta}|\mathbf{y})$ (it is denoted as RIS in Table 8.19); **(g)** another RIS scheme where $f(\boldsymbol{\theta})$ is obtained by a clusterized KDE with $C = 4$ clusters and $h = 0$ (in a similar fashion of Eq. (8.146)); and, finally, a CLAIS scheme with $C \in \{1, 2\}$, $h = 0$ i.e., as in Eq. (8.146).

Table 8.19: Relative MAE, and its corresponding standard error, in estimating the marginal likelihood by seven methods

Methods	Naive	Laplace (soph)	Laplace	HM	RIS	RIS-kde	CLAIS	CLAIS
RE	0.057	0.181	0.553	0.823	0.265	0.140	0.084	0.082
std err	0.001	0.013	0.003	0.018	0.006	0.004	0.015	0.014
comments	—	see [23]	—	—	—	$C = 4$	$C = 1$	$C = 2$

All estimators consider 10000 posterior evaluations. To obtain the samples from the posterior, we run $T = 10000$ iterations of a Metropolis-Hastings algorithm, using the prior as an independent proposal pdf. The IS estimator employs 5000 posterior samples to build the normal approximation to the posterior, from which it draws 5000 additional samples. Similarly, since CLAIS draws additional samples from $\bar{q}(\boldsymbol{\theta})$ in the lower layer, in order to provide a fair comparison, we consider $N = 1$ (i.e. one chain), with $T' = T/2 = 5000$ iterations and sample 5000 additional samples in the lower layer. We averaged the relative MAE over 1000 independent runs. Our results are shown in Table 8.19.

In this example, and with these priors, the results show that the best performing estimator in this case is the Naive Monte Carlo, since prior and likelihood has an ample overlapping region of probability mass. However, the naive Monte Carlo scheme is generally inefficient when there is a small overlap between likelihood and prior. Note also that IS and CLAIS provide good performance. RIS-kde performs better than RIS since the choice of $f(\boldsymbol{\theta})$ in the former is probably narrower than in RIS. The worst performance is provided by the HM estimator.

Scenario 2. Now, we consider the following estimators: **(a)** the Chib's estimator in Eq. (8.30), **(b)** RIS with $f(\boldsymbol{\theta})$ equal to the clusterized KDE in (8.146) (called RIS-kde in the previous scenario), and ((**c**) CLAIS with clusterized KDE in (8.146). We study the ef-

fect of the choice of C and h in their performance. We test different numbers of clusters $C \in \{1, 2, 4, 10\}$ and different values of $h = \{0, 1, 2, 3, 4, 5\}$.
As above, we consider a fair application of CLAIS (using the same budget of posterior evaluations as in the other schemes). Moreover, in Chib's we need to choose the point $\boldsymbol{\theta}^*$. We considered two scenarios: (i) using $\boldsymbol{\theta}^* = [19, 1]$ that is intentionally located very close to the posterior mode; (ii) using random $\boldsymbol{\theta}^*$ drawn from the priors. The first scenario clearly yields more accurate results than the second one, which we refer as a "fair" scenario (since, generally, we do not have information about the posterior modes). In summary, we compute the relative MAE of $\widehat{Z}_{\text{chib}}$, $\widehat{Z}_{\text{chib-f}}$ (where the "f" stands for "fair"), $\widehat{Z}_{\text{RIS}}$ and $\widehat{Z}_{\text{CLAIS}}$. We compute the relative median absolute error of 1000 independent runs. Figure 8.14 shows the results of the experiment. CLAIS and RIS provide results, for all C and h, similar to both Chib and Chib-f. As expected, the error of $\widehat{Z}_{\text{chib}}$ is lower than $\widehat{Z}_{\text{chib-f}}$. In CLAIS, we note that, for $C = 10$, we should not take h too small to avoid the proposal becoming problematic (i.e., narrower than the posterior). Generally, as C increases, h should not be too small since the proposal may not have fatter tails than $P(\boldsymbol{\theta}|\mathbf{y})$. The performance of RIS is best when $h = 0$, and gets worse as h increases, as expected, since $f(\boldsymbol{\theta})$ may become wider than the posterior. We expect that the results of RIS with $h = 0$ would improve further as C increases since the C-KDE pdf, in Eq. (8.146), will have lighter tails than the posterior. The Chib's estimator provides also robust and good results. Overall, for the choices of C and h considered, CLAIS and RIS (with $f(\boldsymbol{\theta})$ being the clusterized KDE) provide robust results comparable to Chib's estimator. These results are also in line with the theoretical considerations given in the Suppl. Material regarding RIS and IS.

8.8.5. Experiment with COVID-19 data

Let us consider data $\mathbf{y} = [y_1, \ldots, y_{D_y}]^\top$ representing the number of daily deaths caused by SAR-CoV-2 in Italy from 18 February 2020 to 6 July 2020. Let t_i denote the i-th day, we model the each observation as

$$y_i = f(t_i) + e_i, \quad i = 1, \ldots, D_y = 140,$$

where f is the function that we aim to approximate and e_i's are Gaussian perturbations. We consider the approximation of f at some t as a weighted sum of M localized basis functions,

$$f(t) = \sum_{m=1}^{M} \rho_m \psi(t|\mu_m, h, \nu),$$

where $\psi(t|\mu_m, h)$ is m-th basis centered at μ_m with bandwidth h. Let also be ν an index denoting the type of basis. We consider $M \in \{1, ..., D_y\}$, then $M \leq D_y$. When $M = D_y$, the model becomes a Relevance Vector Machine (RVM), and the interpolation of all data points (maximum overfitting, with zero fitting error) is possible [71].
We consider 4 different types of basis (i.e., $\nu = 1, ..., 4$): Gaussian ($\nu = 1$), Laplacian

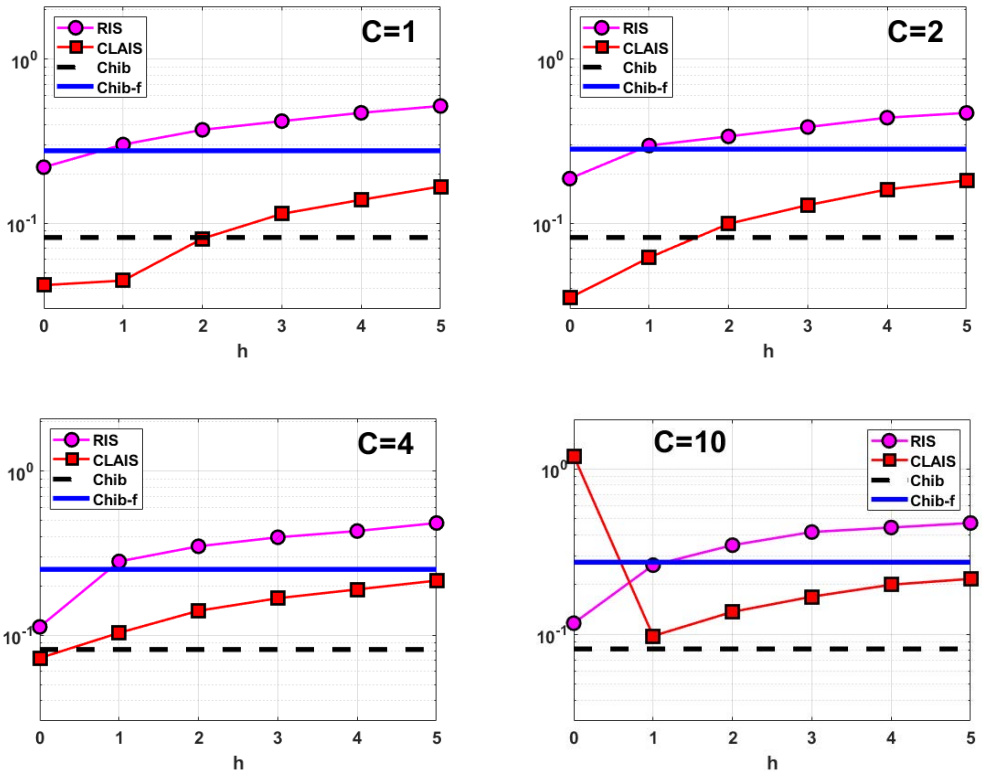


Figure 8.14: Relative median absolute error of RIS and CLAIS versus h for $C \in \{1, 2, 4, 10\}$. The horizontal lines correspond to Chib's estimator (dashed) and its fair application (solid).

($\nu = 2$), Rectangular ($\nu = 3$) and Triangular-Pyramidal ($\nu = 4$). Given ν and M, we select the locations $\{\mu_m\}_{m=1}^M$ as a uniform grid in the interval $[1, D_y]$ (recall that $D_y = 140$). Hence, knowing ν and M, the locations $\{\mu_m\}_{m=1}^M$ are given.

Likelihood and prior of $\boldsymbol{\rho}$. Let $\boldsymbol{\Psi}$ be a $D_y \times M$ matrix with elements $[\boldsymbol{\Psi}]_{i,m} = \psi(t_i|\mu_m, h)$ for $i = 1, \ldots, D_y$ and $m = 1, \ldots, M$, and let $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_M]^\top$ be the vector of coefficients, where M is the total number of bases. Then, the observation equation in vector form becomes

$$\mathbf{y} = \boldsymbol{\Psi}\boldsymbol{\rho} + \mathbf{e},$$

where $\mathbf{e}$ is a $D_y \times 1$ vector of noise. We assume normality $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma_e^2 \mathbf{I}_{D_y})$, where $\mathbf{I}_{D_y}$ is the $D_y \times D_y$ identity matrix. Therefore, the likelihood function is $\ell(\mathbf{y}|\boldsymbol{\rho}, h, \sigma_e, \nu, M) = \mathcal{N}(\mathbf{y}|\boldsymbol{\Psi}\boldsymbol{\rho}, \sigma_e^2 \mathbf{I}_N)$. We also consider a Gaussian prior density over the vector of coefficients $\boldsymbol{\rho}$, i.e., $g(\boldsymbol{\rho}|\lambda) = \mathcal{N}(\boldsymbol{\rho}|\mathbf{0}, \boldsymbol{\Sigma}_\rho)$. where $\boldsymbol{\Sigma}_\rho = \lambda \mathbf{I}_M$ and $\lambda > 0$. Given ν, M, h and σ_e. Thus, the complete set of parameters is $\{\boldsymbol{\rho}, \nu, M, h, \lambda, \sigma_e\}$.

Posteriors and marginalization. With our choice of $g(\boldsymbol{\rho}|\lambda)$, the posterior of $\boldsymbol{\rho}|\lambda, h, \sigma_e$ is also Gaussian,

$$P(\boldsymbol{\rho}|\mathbf{y}, \lambda, h, \sigma_e, \nu, M) = \frac{\ell(\mathbf{y}|\boldsymbol{\rho}, h, \sigma_e, \nu, M) g(\boldsymbol{\rho}|\lambda)}{p(\mathbf{y}|\lambda, h, \sigma_e, \nu, M)} = \mathcal{N}(\boldsymbol{\rho}|\boldsymbol{\mu}_{\rho|y}, \boldsymbol{\Sigma}_{\rho|y}),$$

and a likelihood marginalized w.r.t. $\boldsymbol{\rho}$ is available in closed-form,

$$p(\mathbf{y}|\lambda, h, \sigma_e, \nu, M) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \boldsymbol{\Psi}\boldsymbol{\Sigma}_\rho\boldsymbol{\Psi}^\top + \sigma_e^2\mathbf{I}_N). \tag{8.206}$$

For further details see [71]. Now, we consider priors over h, λ, σ_e, and study the following posterior

$$P(\lambda, h, \sigma_e|\mathbf{y}, \nu, M) = \frac{1}{p(\mathbf{y}|\nu, M)} p(\mathbf{y}|\lambda, h, \sigma_e, \nu, M) g_\lambda(\lambda) g_h(h) g_\sigma(\sigma_e),$$

where $g_\lambda(\lambda)$, $g_h(h)$, $g_\sigma(\sigma_e)$ are folded-Gaussian pdfs defined on $\mathbb{R}_+ = (0, \infty)$ with location and scale parameters $\{0, 100\}$, $\{0, 400\}$ and $\{1.5, 9\}$, respectively. Finally, we want to compute the marginal likelihood of this posterior, i.e.,

$$p(\mathbf{y}|\nu, M) = \int_{\mathbb{R}_+^3} p(\mathbf{y}|\lambda, h, \sigma_e, \nu, M) g_\lambda(\lambda) g_h(h) g_\sigma(\sigma_e) d\lambda dh d\sigma_e. \tag{8.207}$$

Furthermore, assuming a uniform probability mass $\frac{1}{D_y}$ as prior over M, we can also marginalize out M,

$$p(M|\mathbf{y}, \nu) \propto \frac{1}{D_y} p(\mathbf{y}|\nu, M) \quad \text{and} \quad p(\mathbf{y}|\nu) = \frac{1}{D_y} \sum_{M=1}^{D_y} p(\mathbf{y}|\nu, M), \quad \text{for} \quad \nu = 1, ..., 4. \tag{8.208}$$

Considering also a uniform prior over ν, we can obtain $p(\nu|\mathbf{y}) \propto \frac{1}{4} p(\mathbf{y}|\nu)$.

For approximating $p(\mathbf{y}|\nu, M)$, for $m = 1, \ldots, D_y$, we first apply a Naive Monte Carlo (NMC) method with $N = 10^4$ samples. Secondly, we run an MTM algorithm for obtaining the estimator $\widehat{Z}^{(2)}$ (see Table 8.13) and a Markov chain of vectors $\boldsymbol{\theta}_t = [\lambda_t, h_t, \sigma_{e,t}]$ for $t = 1, \ldots, T$. This generated chain $\{\boldsymbol{\theta}_t\}_{t=1}^T$ can be also used for obtaining other estimators (e.g., the HM estimator). We consider the pairs $T = 50$, $N' = 1000$, in the MTM scheme. Therefore, $\widehat{Z}^{(2)}$ employs $N'T = 5 \cdot 10^4$ samples.

Goal. Our purpose is: (a) to make inference regarding the parameters of the model $\{\lambda, h, \sigma_e\}$, (b) approximate $Z = p(\mathbf{y}|\nu, M)$, (c) study the posterior $p(M|\mathbf{y}, \nu)$, and (d) obtain the MAP value, M_ν^*, for $\nu = 1, ..., 4$. We also study the marginal posterior $p(\nu|\mathbf{y})$ of each of the four candidate bases.

Results. We run once NMC and MTM for all $M = 1, ..., D_y = 140$ different models and approximate the posterior $p(M|\mathbf{y}, \nu)$ for each value of M. For illustrative reasons, in Figure 8.15, we show the posterior probabilities of M belonging to the intervals $[4\widetilde{M}-3, 4\widetilde{M}]$, where $\widetilde{M}$ is an auxiliary index $\widetilde{M} = 1, ..., \frac{140}{4} = 35$. Thus, the first value, $\widetilde{M} = 1$ of the curves in Figure 8.15, represents the probability of $M \in \{1, 2, 3, 4\}$, the second value represents the probability of $M \in \{5, 6, 7, 8\}$, and so on until the last value, $\widetilde{M} = 35$, which represents the probability of $M \in \{137, 138, 139, 140\}$. We can observe that, with both techniques, we obtain that $\widetilde{M} = 2$ is the most probable interval, with a probability generally closer to 0.2, hence $M_\nu^* \in \{5, 6, 7, 8\}$. Recall that we have 35 possible intervals (values of $\widetilde{M}$), so when we compare with a uniform distribution $\frac{1}{35} = 0.0286$, the value 0.2 is quite high. For $\nu = 2, 3$, the corresponding probabilities are greater than 0.2, reaching 0.35 with

NMC in $\nu = 2$. In Figure 8.16, we can observe that, with $M = 8$ bases, we are already able to obtain a very good fitting to the data.

Thus, a first conclusion is that the results obtained with models such as RVMs and Gaussian Processes (GPs) (both having $M = 140$ [71]) can be approximated in a very good way with a much more scalable model, as our model here with $M \in \{5, 6, 7, 8\}$ [71]. Regarding the marginal posterior $p(\nu|\mathbf{y})$, we can observe the results in Table 8.20. The basis $\nu = 3$ is discarded since is clearly not appropriate, as also shown graphically by Figure 8.16. With the results provided by NMC, we prefer slightly the Laplacian basis whereas, with the results of MTM, we have almost $p(\nu = 1|\mathbf{y}) \approx p(\nu = 2|\mathbf{y})$. These considerations are reasonable after having a look to Figure 8.16. As future work, it would be interesting to consider the locations of the bases μ_m, for $m = 1, \ldots, M$, as additional parameters to be learnt.

Table 8.20: The approximate marginal posterior $p(\nu|\mathbf{y})$ with different techniques.

Method	Number of used samples	$p(\nu = 1\|\mathbf{y})$	$p(\nu = 2\|\mathbf{y})$	$p(\nu = 3\|\mathbf{y})$	$p(\nu = 4\|\mathbf{y})$
NMC	10^4	0.3091	0.3307	0.0813	0.2790
MTM	$5 \cdot 10^4$	0.3155	0.3100	0.0884	0.2861

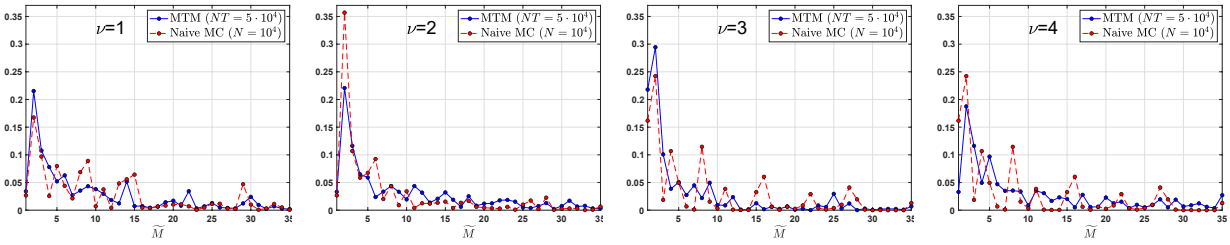


Figure 8.15: Posterior probabilities of the intervals $[4\widetilde{M} - 3, 4\widetilde{M}]$ with $\widetilde{M} = 1, ..., 35$, obtained adding 4 consecutive values of $p(M|\mathbf{y}, \nu)$ with $M \in \{4\widetilde{M} - 3, 4\widetilde{M} - 2, 4\widetilde{M} - 1, 4\widetilde{M}\}$ (and $p(M|\mathbf{y}, \nu)$ is approximated by NMC or MTM). Each figure corresponds to a different type of basis, $\nu = 1, 2, 3, 4$.

8.9. Final discussion

In this work, we have provided an exhaustive review of the techniques for marginal likelihood computation with the purpose of model selection and hypothesis testing. Methods for approximating ratios of normalizing constants have been also described. The relationships among all of them have been widely described in the text, for instance in Sections 8.4.2 and 8.4.3, by means of several summary tables (see, as examples, Tables 8.5, 8.8, and 8.16) and Figures from 8.1 to 8.6. The careful choice of the prior and the careful use of the improper priors in the Bayesian setting have been discussed. A brief description of alternative model selection strategies based on the posterior predictive approach, has been also provided.

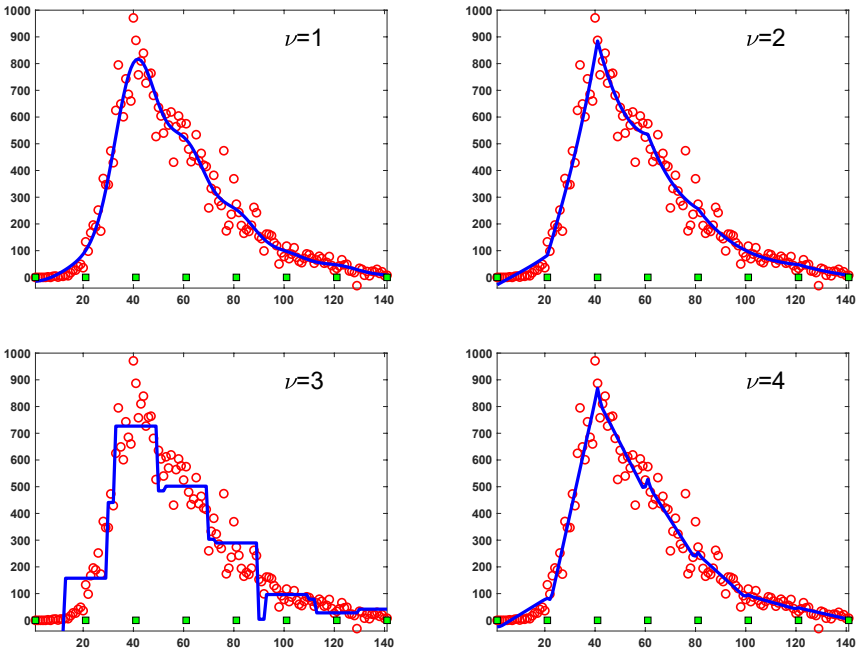


Figure 8.16: Best fit with 8 bases with different types of basis, $\nu = 1, 2, 3, 4$. The circles represent the analyzed data and the squares show the positions of the bases.

Most of the presented computational techniques are based on the importance sampling (IS) approach, but also require the use of MCMC algorithms. Table 8.21 summarizes some methods for estimating Z, which involve the generation of the posterior $P(\boldsymbol{\theta}|\mathbf{y})$ (without using other tempered versions). This table is devoted to the interested readers which desire to obtain samples $\{\boldsymbol{\theta}_n\}_{n=1}^N$ by an MCMC method with invariant pdf $P(\boldsymbol{\theta}|\mathbf{y})$ (without either any tempering or sequence of densities) and, at the same time, also desire to approximate Z. Clearly, this table provides only a subset of all the possible techniques. They can be considered the simplest schemes, in the sense that they do not use any tempering strategy or sequence of densities. We also recall that AIC and DIC are commonly used for model comparison, although they do not directly target the actual marginal likelihood. Table 8.22 enumerates all the schemes that require the sampling and evaluation of tempered posteriors. For LAIS, the use of tempered posteriors is not strictly required. In PS, one could select a path that does not involve tempered posteriors. The schemes which provides unbiased estimators of Z or $\log Z$ are given in Table 8.23.
We also provide some final advice for practical use. First of all, if informative priors are not available, a very careful choice of the priors must be considered, as remarked in Sections 8.7.1 and 8.7.2, or alternatively a predictive posterior approach should be applied (see Section 8.7.4). From a computational point of view, our suggestions are listed below:

- The use of Naive Monte Carlo (NMC) should be always considered, at least as a first attempt. Moreover, the HM estimator is surely the worst estimator of Z, but it could be applied for obtaining an upper bound for Z, although it can be very imprecise/loose.

- The application of an MTM method is a good choice within the MCMC schemes. In fact, as shown in Figure 8.17, it also provides two estimators of Z, as well as a set of samples. These samples can be employed in other estimators, including Chib, RIS and LAIS, for instance.

- Regarding the more general task of estimating ratio of constants, In [13], the authors show that (given two unnormalized pdfs) the optimal umbrella estimator provides the best performance theoretically in estimating the ratio of their normalizing constants. However, the optimal umbrella sampling estimator is difficult and costly to implement (due to the fact sampling from the optimal umbrella proposal is not straightforward), so its best performance may not be achieved in practice.

- The Chib's method is a good alternative, that provide very good performance as we can observe in Section 8.8.4 and also in [58, 32]. Moreover, the Chib's method is also related to bridge sampling as discussed in Section 8.4.2. However, since it requires internal information regarding the MCMC employed (proposal, acceptance function etc.), it cannot be considered for a possible post-processing scheme after obtaining a Markov chain from a black-box MCMC algorithm. This could be easily done with the HM estimator or LAIS, for instance.

- LAIS can be considered a scheme in between the NMC and HM. NMC draw samples from the prior, which makes it rather inefficient in some setting. The HM estimator uses posterior samples but it is very unstable. LAIS uses the posterior samples to build a suitable normalized proposal, so it benefits from localizing samples in regions of high posterior probability (like the HM), while preserving the properties of standard IS (like the Naive MC). In this sense, bridge sampling, the SS method, path sampling, and the rest of techniques based on tempered posteriors, are also schemes in between the NMC and HM.

- The methods based on tempered posteriors provide very good performance but the choice of the temperature parameters β_k is important. In our opinion, among SS, PS, PP, An-IS, and SMC, the more robust to the choice of the β_k's is the SS method (that is, perhaps, also the simplest one). Moreover, The SS method does not require the use of several tempered posteriors, unlike PS and PP. The LAIS technique can also be employed in the upper layer. Since the samples in the upper layer are only used as means of other proposal pdfs and, in the lower layer, the true posterior $P(\boldsymbol{\theta}|\mathbf{y})$ is always evaluated, LAIS is also quite robust to the choice of β_k. More comparisons among SS, An-IS, and SMC are required, since these methods are also very related as depicted in Figures 8.2, 8.4, 8.5 and 8.6.

- The nested sampling technique has gained attention and is largely applied in the literature. The derivation is complex and several approximations are considered, as discussed in Section 8.6.2. The sampling from the truncated priors is the key point and it is not straightforward [18]. In this sense, its success in the literature is

surprising. However, the nested sampling includes an implicit optimization of the likelihood. We believe that is an important feature, since the knowledge of high probabilities of the likelihood is a crucial point also to the rest of computational schemes.

Table 8.21: Schemes for estimating Z, involving MCMC samples from $P(\boldsymbol{\theta}|\mathbf{y})$.

Method	**Section**	**Need of drawing additional samples**	**Comments**
Below: methods for **post-processing** after generating N MCMC samples from $P(\boldsymbol{\theta}|\mathbf{y})$.			
Laplace	8.3.1	—	use MCMC for estimating $\widehat{\boldsymbol{\theta}}_{\text{MAP}}$
BIC	8.3.2	—	use MCMC for estimating $\widehat{\boldsymbol{\theta}}_{\text{MLE}}$
KDE	8.3.3	—	use MCMC for generating samples
Bridge	8.4.2	✓	additional samples are required; see Eq. (8.61)
RIS	8.4	—	the HM estimator is a special case
MTM	8.5.3	—	provides two estimators of Z
LAIS	8.5.4	✓	with $P(\boldsymbol{\theta}|\mathbf{y})$ in the upper-layer
Below: methods that require internal information of the MCMC scheme.			
Chib's method	8.3.4	✓	additional samples are required if the proposal is not independent
MTM	8.5.3	—	provides two estimators of Z
Below: for model selection but do not approximate the marginal likelihood			
AIC	8.3.2	—	use MCMC for estimating $\widehat{\boldsymbol{\theta}}_{\text{MLE}}$
DIC	8.3.2	—	use MCMC for estimating c_p and $\bar{\boldsymbol{\theta}}$

Table 8.22: Methods using tempered posteriors.

Method	**Section**	**Use of tempering strictly required**
IS-P	8.4.3	without tempering, it is HM
Stepping Stones (SS)	8.4.3	✓
Path Sampling (PS)	8.4.3	other paths (without tempering) can be used
Method of Power Posteriors (PP)	8.4.3	✓
Annealed Importance Sampling (An-IS)	8.5.1	✓
Sequential Monte Carlo (SMC)	8.5.2	✓
Layered Adaptive Importance Sampling (LAIS)	8.5.4	—

Table 8.23: Methods providing unbiased estimators of Z or $\log Z$.

Method	Section
Unbiased estimators of Z:	
IS vers-1	8.4
Stepping Stones (SS)	8.4.3
Annealed Importance Sampling (An-IS)	8.5.1
Sequential Monte Carlo (SMC)	8.5.2
Layered Adaptive Importance Sampling (LAIS)	8.5.4
Unbiased estimators of $\log Z$:	
Path Sampling (PS)	8.4.3

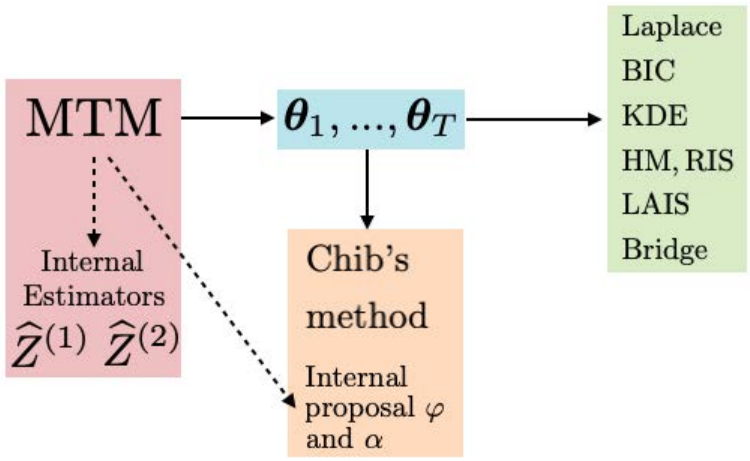


Figure 8.17: The application of the MTM algorithm as MCMC provides the generated samples $\{\boldsymbol{\theta}_1, \ldots \boldsymbol{\theta}_T\}$ and also two possible estimators of Z. The generated samples can employed in other schemes including RIS, LAIS and Bridge sampling. Moreover, considering the proposal and the acceptance function α of the MTM, the Chib's method can be also applied. Indeed, the MTM yields a reversible chain (i.e., fulfills the balance condition).

Appendix

Table of other reviews

The related literature is rather vast. In this section, we provide a brief summary that intends to be illustrative rather than exhaustive, by means of Table 8.24. The most relevant (in our opinion) and related surveys are compared according to the topics, material and schemes described in lthe work. The proportion of covering and overlapping with this work is roughly classified as "partial" $\diamond$, "complete" $\surd$, "remarkable" or "more exhaustive" work with $\bigstar$. From Table 8.24, we can also notice the completeness of this work. We take into account also the completeness and the depth of details provided in the different derivations. The Christian Robert's blog deserves a special mention (https://xianblog.wordpress.com), since Professor C. Robert has devoted several

entries of his blog with very interesting comments regarding the marginal likelihood estimation and related topics.

Table 8.24: Covering of the considered topics of other surveys or works (◇: partial, √: complete, ★: remarkable or more exhaustive). We take into account also the completeness and the depth of details provided in the different derivations. To be more precise, in the case of Section 4.1, we have also considered the subsections.

Surveys	Families 1–2	IS			Advanced schemes						Vertical likelihood			Improper
		1 prop.	2 prop.	Multiple	MCMC within IS			MTM	AIS		5.1	5.2	5.3	
Gelfand and Dey (1994)[33]	✓	◇												
Kass and Raftery (1995)[47]														
Raftery (1995)[39, Ch. 10]	◇	✓	◇											
Meng and Wong (1996)[74]		◇	★	◇										
DiCiccio et al (1997)[23]	★	✓	✓											
Chen and Shao (1997)[13]	◇	◇	✓	✓										
Chen et al (2012)[14, Ch. 5]														
Gelman and Meng (1997)[35]		✓	★	★										
Bos (2002)[5]	✓	✓												
Vyshemirsky and Girolami (2007)[103]		◇		◇		◇								
Marin and Robert (2009)[58]	◇	✓	◇											
Robert and Wraith (2009)[89]		✓	✓									✓		
Friel and Wyse (2012)[32]	◇	◇		◇		✓						✓		
Ardia et al (2012)[2]	◇	◇	✓											
Polson and Scott (2014)[84]		◇		◇							✓	★	★	
Schöniger et al (2014)[92]	✓	✓										◇		
Knuth et al (2015)[48]	◇	◇		✓		✓						✓		
Liu et al (2016)[55]	◇	◇		✓								★		
Zhao and Severini (2017)[107]	★	✓	✓											
Martino (2018)[59]								◇	★					
Bugallo et al (2017)[7]		◇								★				
Bugallo et al (2015)[8]														
Oaks et al (2019)[86]		◇		✓		◇	◇					◇		
O'Hagan (1995)[81]														★
Berger and Pericchi (1996)[4]														

Bibliography

[1] C. Alston, P. Kuhnert, L. S. Choy, R. McVinish, and K. Mengersen. Bayesian model comparison: Review and discussion. *International Statistical Insitute, 55th session*, 2005.

[2] D. Ardia, N. Baştürk, L. Hoogerheide, and H. K. Van Dijk. A comparative study of Monte Carlo methods for efficient evaluation of marginal likelihood. *Computational Statistics & Data Analysis*, 56(11):3398–3414, 2012.

[3] V. Balasubramanian. Statistical inference, Occam's razor, and statistical mechanics on the space of probability distributions. *Neural computation*, 9(2):349–368, 1997.

[4] J. O. Berger and L. R. Pericchi. The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122, 1996.

[5] C. S. Bos. A comparison of marginal likelihood computation methods. In *Compstat*, pages 111–116. Springer, 2002.

[6] F.-X. Briol, C. J. Oates, M. Girolami, M. A. Osborne, and D. Sejdinovic. Probabilistic integration: A role in statistical computation? *Statistical Science*, 34(1):1–22, 2019.

[7] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric. Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017.

[8] M. F. Bugallo, L. Martino, and J. Corander. Adaptive importance sampling in signal processing. *Digital Signal Processing*, 47:36–49, 2015.

[9] E. Cameron and A. Pettitt. Recursive pathways to marginal likelihood estimation with prior-sensitivity analysis. *Statistical Science*, 29(3):397–419, 2014.

[10] O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.

[11] B. P. Carlin and S. Chib. Bayesian model choice via Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(3):473–484, 1995.

[12] M.-H. Chen. Importance-weighted marginal Bayesian posterior density estimation. *Journal of the American Statistical Association*, 89(427):818–824, 1994.

[13] M.-H. Chen, Q.-M. Shao, et al. On Monte Carlo methods for estimating ratios of normalizing constants. *The Annals of Statistics*, 25(4):1563–1594, 1997.

[14] M. H. Chen, Q. M. Shao, and J. G. Ibrahim. *Monte Carlo methods in Bayesian computation*. Springer, 2012.

[15] S. Chib. Marginal likelihood from the Gibbs output. *Journal of the american statistical association*, 90(432):1313–1321, 1995.

[16] S. Chib and I. Jeliazkov. Marginal likelihood from the Metropolis–Hastings output. *Journal of the American Statistical Association*, 96(453):270–281, 2001.

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

[18] N. Chopin and C. P. Robert. Properties of nested sampling. *Biometrika*, 97(3):741–755, 2010.

[19] G. Claeskens and N. L. Hjort. The focused information criterion. *Journal of the American Statistical Association*, 98(464):900–916, 2003.

[20] P. Congdon. Bayesian model choice based on Monte Carlo estimates of posterior model probabilities. *Computational statistics & data analysis*, 50(2):346–357, 2006.

[21] J. M. Cornuet, J. M. Marin, A. Mira, and C. P. Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, December 2012.

[22] P. Dellaportas, J. J. Forster, and I. Ntzoufras. On Bayesian model and variable selection using MCMC. *Statistics and Computing*, 12(1):27–36, 2002.

[23] T. J. DiCiccio, R. E. Kass, A. Raftery, and L. Wasserman. Computing Bayes factors by combining simulation and asymptotic approximations. *Journal of the American Statistical Association*, 92(439):903–915, 1997.

[24] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, September 2003.

[25] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. *technical report*, 2008.

[26] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Efficient multiple importance sampling estimators. *IEEE Signal Processing Letters*, 22(10):1757–1761, 2015.

[27] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Heretical multiple importance sampling. *IEEE Signal Processing Letters*, 23(10):1474–1478, 2016.

[28] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Generalized Multiple Importance Sampling. *Statistical Science*, 34(1):129–155, 2019.

[29] E. Fong and C.C. Holmes. On the marginal likelihood and cross-validation. *Biometrika*, 107(2):489–496, 2020.

[30] N. Friel, M. Hurn, and J. Wyse. Improving power posterior estimation of statistical evidence. *Statistics and Computing*, 24(5):709–723, 2014.

[31] N. Friel and A. N. Pettitt. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):589–607, 2008.

[32] N. Friel and J. Wyse. Estimating the evidence-a review. *Statistica Neerlandica*, 66(3):288–308, 2012.

[33] A. E. Gelfand and D. K. Dey. Bayesian model choice: asymptotics and exact calculations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(3):501–514, 1994.

[34] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data analysis*. CRC press, 2013.

[35] A. Gelman and X. L. Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical science*, pages 163–185, 1998.

[36] C. J. Geyer. Estimating normalizing constants and reweighting mixtures. *Technical Report, number 568 - School of Statistics, University of Minnesota*, 1994.

[37] W. R. Gilks and C. Berzuini. Following a moving target-Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(1):127–146, 2001.

[38] W. R. Gilks, N. G. Best, and K. K. C. Tan. Adaptive Rejection Metropolis Sampling within Gibbs Sampling. *Applied Statistics*, 44(4):455–472, 1995.

[39] W. R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC, 1995.

[40] W. R. Gilks and P. Wild. Adaptive Rejection Sampling for Gibbs Sampling. *Applied Statistics*, 41(2):337–348, 1992.

[41] S. J. Godsill. On the relationship between Markov chain Monte Carlo methods for model uncertainty. *Journal of computational and graphical statistics*, 10(2):230–248, 2001.

[42] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[43] P. Grunwald and T. Roos. Minimum Description Length Revisited. *arXiv:1908.08484*, pages 1–38, 2019.

[44] E. J. Hannan and B. G. Quinn. The determination of the order of an autoregression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2):190–195, 1979.

[45] D. I. Hastie and P. J. Green. Model choice using reversible jump Markov chain Monte Carlo. *Statistica Neerlandica*, 66(3):309–338, 2012.

[46] J. A. Hoeting, D. Madigan, A. E. Raftery, and Chris T. Volinsky. Bayesian model averaging: a tutorial. *Statistical Science*, 14(4):382–417, 1999.

[47] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[48] K. H. Knuth, M. Habeck, N. K. Malakar, A. M. Mubeen, and B. Placek. Bayesian evidence and model selection. *Digital Signal Processing*, 47:50–67, 2015.

[49] A. Kong. A note on importance sampling using standardized weights. *Technical Report 348, Department of Statistics, University of Chicago*, 1992.

[50] S. Konishi and G. Kitagawa. *Information criteria and statistical modeling*. Springer Science & Business Media, 2008.

[51] C. H. LaMont and P. A. Wiggins. Correspondence between thermodynamics and inference. *Physical Review E*, 99(5):052140, 2019.

[52] S. M. Lewis and A. E. Raftery. Estimating Bayes factors via posterior simulation with the Laplace—Metropolis estimator. *Journal of the American Statistical Association*, 92(438):648–655, 1997.

[53] F. Liang, C. Liu, and R. Caroll. *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*. Wiley Series in Computational Statistics, England, 2010.

[54] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2004.

[55] P.i Liu, A. S. Elshall, M. Ye, P. Beerli, X. Zeng, D. Lu, and Y. Tao. Evaluating marginal likelihood with thermodynamic integration method and comparison with several other numerical methods. *Water Resources Research*, 52(2):734–758, 2016.

[56] Q. Liu, J. Peng, A. Ihler, and J. Fisher III. Estimating the partition function by discriminance sampling. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 514–522, 2015.

[57] D. Luengo, L. Martino, M. Bugallo, V. Elvira, and Sarkka S. A survey of monte carlo methods for parameter estimation. *EURASIP J. Adv. Signal Process.*, 25:1–62, 2020.

[58] J. M. Marin and C. P. Robert. Importance sampling methods for Bayesian discrimination between embedded models. *arXiv preprint arXiv:0910.2325*, 2009.

[59] L. Martino. A review of multiple try MCMC algorithms for signal processing. *Digital Signal Processing*, 75:134 – 152, 2018.

[60] L. Martino, R. Casarin, F. Leisen, and D. Luengo. Adaptive independent sticky MCMC algorithms. *EURASIP Journal on Advances in Signal Processing (to paper)*, 2017.

[61] L. Martino and V. Elvira. Metropolis sampling. *Wiley StatsRef: Statistics Reference Online*, pages 1–18, 2017.

[62] L. Martino and V. Elvira. Compressed Monte Carlo for distributed Bayesian inference. *viXra:1811.0505*, 2018.

[63] L. Martino, V. Elvira, and G. Camps-Valls. Group importance sampling for particle filtering and MCMC. *Digital Signal Processing*, 82:133 – 151, 2018.

[64] L. Martino, V. Elvira, and F. Louzada. Weighting a resampled particle in Sequential Monte Carlo. *IEEE Statistical Signal Processing Workshop, (SSP)*, 122:1–5, 2016.

[65] L. Martino, V. Elvira, and M. F. Louzada. Effective Sample Size for importance sampling based on the discrepancy measures. *Signal Processing*, 131:386–401, 2017.

[66] L. Martino, V. Elvira, and D. Luengo. Anti-tempered layered adaptive importance sampling. *International Conference on Digital Signal Processing (DSP)*, 2017.

[67] L. Martino, V. Elvira, D. Luengo, and J. Corander. Layered adaptive importance sampling. *Statistics and Computing*, 27(3):599–623, 2017.

[68] L. Martino, D. Luengo, and J. Míguez. Independent random sampling methods. *Springer*, 2018.

[69] L. Martino, V. P. Del Olmo, and J. Read. A multi-point Metropolis scheme with generic weight functions. *Statistics & Probability Letters*, 82(7):1445–1453, 2012.

[70] L. Martino and J. Read. On the flexibility of the design of multiple try Metropolis schemes. *Computational Statistics*, 28(6):2797–2823, 2013.

[71] L. Martino and J. Read. Joint introduction to Gaussian Processes and Relevance Vector Machines with connections to Kalman filtering and other kernel smoothers. *arXiv:2009.09217*, pages 1–50, 2020.

[72] L. Martino, J. Read, V. Elvira, and F. Louzada. Cooperative parallel particle filters for on-line model selection and applications to urban mobility. *Digital Signal Processing*, 60:172–185, 2017.

[73] L. Martino, J. Read, and D. Luengo. Independent doubly adaptive rejection Metropolis sampling within Gibbs sampling. *IEEE Transactions on Signal Processing*, 63(12):3123–3138, June 2015.

[74] X.-L. Meng and W. H. Wong. Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, pages 831–860, 1996.

[75] A. Mira and G. Nicholls. Bridge estimation of the probability density at a point. Technical report, Department of Mathematics, The University of Auckland, New Zealand, 2003.

[76] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[77] C. A. Naesseth, F. Lindsten, and T. B. Schon. Nested Sequential Monte Carlo methods. *Proceedings of theInternational Conference on Machine Learning*, 37:1–10, 2015.

[78] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

[79] M. A. Newton and A. E. Raftery. Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(1):3–26, 1994.

[80] C. J. Oates, T. Papamarkou, and M. Girolami. The controlled thermodynamic integral for Bayesian model evidence evaluation. *Journal of the American Statistical Association*, 111(514):634–645, 2016.

[81] A. O'Hagan. Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):99–118, 1995.

[82] R. B. O'Hara and M. J. Sillanpää. A review of Bayesian variable selection methods: what, how and which. *Bayesian analysis*, 4(1):85–117, 2009.

[83] J. Piironen and A. Vehtari. Comparison of Bayesian predictive methods for model selection. *Statistics and Computing*, 27(3):711–735, 2017.

[84] N. G. Polson and J. G. Scott. Vertical-likelihood Monte Carlo. *arXiv preprint arXiv:1409.3601*, 2014.

[85] C. M. Pooley and G. Marion. Bayesian model evidence as a practical alternative to deviance information criterion. *Royal Society Open Science*, 5(3):1–16, 2018.

[86] J. R Oaks, K. A. Cobb, V. N Minin, and A. D. Leaché. Marginal likelihoods in phylogenetics: a review of methods and applications. *Systematic biology*, 68(5):681–697, 2019.

[87] C. E. Rasmussen and Z. Ghahramani. Bayesian Monte Carlo. *Advances in neural information processing systems*, pages 505–512, 2003.

[88] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[89] C. P. Robert and D. Wraith. Computational methods for Bayesian model choice. *AIP conference proceedings*, 1193(1):251–262, 2009.

[90] H. Rue, A. Riebler, S. H. Sørbye, J. B. Illian, D. P. Simpson, and F. K. Lindgren. Bayesian computing with INLA: a review. *Annual Review of Statistics and Its Application*, 4:395–421, 2017.

[91] Y. Sakamoto, M. Ishiguro, and G. Kitagawa. Akaike information criterion statistics. *Dordrecht, The Netherlands: D. Reidel*, 81, 1986.

[92] A.i Schöniger, T. Wöhling, L. Samaniego, and W. Nowak. Model selection on solid ground: Rigorous comparison of nine ways to evaluate Bayesian model evidence. *Water resources research*, 50(12):9484–9513, 2014.

[93] I. Schuster and I. Klebanov. Markov Chain Importance Sampling-a highly efficient estimator for MCMC. *arXiv preprint arXiv:1805.07179*, 2018.

[94] G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[95] J. Skilling. Nested sampling for general Bayesian computation. *Bayesian analysis*, 1(4):833–859, 2006.

[96] D. J. Spiegelhalter and A. F.M. Smith. Bayes factors for linear and log-linear models with vague prior information. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(3):377–387, 1982.

[97] D.J. Spiegelhalter, N. G. Best, B. P. Carlin, and A. Van der Linde. Bayesian measures of model complexity and fit. *J. R. Stat. Soc. B*, 64:583–616, 2002.

[98] D.J. Spiegelhalter, N. G. Best, B. P. Carlin, and A. Van der Linde. The deviance information criterion: 12 years on. *J. R. Stat. Soc. B*, 76:485–493, 2014.

[99] G.l Stoltz and M. Rousset. Free energy computations: A mathematical perspective. *World Scientific*, 2010.

[100] G. M. Torrie and J. P. Valleau. Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187–199, 1977.

[101] I. Urteaga, M. F. Bugallo, and P. M. Djurić. Sequential Monte Carlo methods under model uncertainty. In *2016 IEEE Statistical Signal Processing Workshop (SSP)*, pages 1–5, 2016.

[102] A. Vehtari, A. Gelman, and J. Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and computing*, 27(5):1413–1432, 2017.

[103] V. Vyshemirsky and M. A. Girolami. Bayesian ranking of biochemical system models. *Bioinformatics*, 24(6):833–839, 2007.

[104] Y.-B. Wang, M.-H. Chen, L. Kuo, and P. O. Lewis. A new Monte Carlo method for estimating marginal likelihoods. *Bayesian analysis*, 13(2):311, 2018.

[105] M. D. Weinberg et al. Computing the Bayes factor from a Markov chain Monte Carlo simulation of the posterior distribution. *Bayesian Analysis*, 7(3):737–770, 2012.

[106] W. Xie, P. O. Lewis, Y. Fan, L. Kuo, and M. H. Chen. Improving marginal likelihood estimation for Bayesian phylogenetic model selection. *Systematic biology*, 60(2):150–160, 2010.

[107] Z. Zhao and T. A. Severini. Integrated likelihood computation methods. *Computational Statistics*, 32(1):281–313, 2017.

9. CONCLUSIONS

To conclude this thesis, we give a summary of the main contributions and discuss potential lines of further research. This thesis has investigated Monte Carlo algorithms for the application of Bayesian inference. The proposed algorithms address the challenges arising in real applications such as highly-concentrated posteriors, multimodality, and posteriors that are very costly to evaluate or come in the form of noisy realizations. We have also addressed the design of efficient proposal distributions. In addition to this, this thesis has made contribution to Bayesian model selection, providing a review of Monte Carlo approaches for marginal likelihood computation, and discussing the prior sensitivity of marginal likelihoods and the use of improper priors.

In order to foster the exploration of the state space, in Chapter 2 and Chapter 3 we have introduced methodologies that make use of population MCMC algorithms, whether independent, interacting or combinations of both. Additionally, the algorithms of Chapter 3 apply multiple importance sampling to the states produced by the population MCMC algorithms (i.e. they are adaptive importance sampling algorithms), which increases even more the exploration and robustness of the final estimators.

In the works presented in Chapter 4, Chapter 5 and Chapter 6, we have considered the use of surrogate models of the posterior, built with nonparametric techniques. The use of surrogate models has been motivated in different, but related ways, since the ultimate goal is to improve the efficiency over default Monte Carlo algorithms. In Chapter 4, the posterior has been assumed to be very costly to evaluate so a surrogate built from a small number of posterior evaluations is used instead. Then, we have proposed applying intensive Monte Carlo, and Gaussian quadratures, to integrals of the surrogate. In Chapter 5, we have considered employing the surrogate as proposal in an adaptive IS algorithm in order to build a really efficient proposal distribution. In Chapter 6, we have provided an unifying view of Monte Carlo algorithms employing surrogates, also in the context of only having access to noisy evaluations of the posterior. Additionally, in Chapter 7 we have studied the setting of noisy IS and obtained expressions for optimal proposals.

Finally, Chapter 8 has addressed the topic of Bayesian model selection. Chapter 8 has provided a comprehensive review of computational approaches for marginal likelihood computation, with special focus on the approaches based on IS.

9.1. Further work

We have identified several lines of further research, which can be summarized as follows.

1. Investigate the trade-off between exploration and exploitation in some of the schemes proposed in this thesis. In the LAIS framework, how many posterior evaluations should we devote to both upper and lower layers. In PMHC, we can formally investigate the trade-off between the number of vertical and horizontal iterations.

2. Although using nonparametric approximations within Monte Carlo has a long history, there is still current development. Namely, the incorporation of surrogate models to reduce the number of posterior evaluations, as well as for the construction more efficient proposal mechanisms, is still motivating new research as proves the recent works [1, 3].

3. In the relation to previous point, a possible research line can be the combination of ideas from two groups of algorithms identified in Chapter 6, namely *iterative refinement* and *exact*. For instance, a biased algorithm can target a surrogate and only apply correction steps when the surrogate is a really bad approximation of the posterior. This would obtain an algorithm that is less costly than a purely exact algorithm and whose bias is decreased faster at the initial iterations, when the surrogate is a crude approximation of the posterior.

4. Although we already discussed the possibility of using more than layer in RADIS, presented in Chapter 5, we can formaly investigate using an adaptive sequence of surrogates with increasing accuracy and cost. This would help both the sampling of the final surrogate thanks to the sequential approach, and also would allow us to reduce costs by stopping in the level where the surrogate is a good enough approximation of the posterior.

5. Instead of running Monte Carlo directly on the posterior, when the posterior is very costly to evaluate it is more appropriate to devote a small number of evaluations to obtain a surrogate model and then obtain a quadrature rule, as presented in Chapter 4. The selection of the nodes is an important consideration that affect the speed of convergence in terms of posterior evaluations. Active learning of the nodes can have a enormous impact on the precision of the quadratures. Provided that specific basis functions produce surrogates with a probabilistic interpretation, the large amount of research from the Bayesian optimization literature can be used for designing much more efficient algorithms. However, proving the convergence of this adaptive Bayesian quadratures could be hard in general, as exposed in [4].

6. In order to extend the use of nonparametric approximations of the posterior to settings where the dimensionality of the parameter space is high, we can employ deep models, such as deep Gaussian Processes [2].

Bibliography

[1] J. J. Bon, A. Lee, and C. Drovandi. Accelerating sequential Monte Carlo with surrogate likelihoods. *Statistics and Computing*, 31(5):1–26, 2021.

[2] A. Damianou and N. D. Lawrence. Deep gaussian processes. In *Artificial intelligence and statistics*, pages 207–215. PMLR, 2013.

[3] A. D. Davis, Y. Marzouk, A. Smith, and N. Pillai. Rate-optimal refinement strategies for local approximation MCMC. *Statistics and Computing*, 32(4):1–23, 2022.

[4] M. Kanagawa, B. K. Sriperumbudur, and K. Fukumizu. Convergence guarantees for kernel-based quadrature rules in misspecified settings. *Advances in Neural Information Processing Systems*, 29, 2016.