Compressed Monte Carlo for Distributed Bayesian Inference

Luca Martino[⊤], Víctor Elvira[*]

[⊤] Dep. of Signal Processing, Universidad Carlos III de Madrid (UC3M)

[*] IMT Lille Douai, Cité Scientifique, Rue Guglielmo Marconi, 20145, Villeneuve dAscq 59653, (France)

***Abstract*—Bayesian models have become very popular over the last years in several fields such as statistics, signal processing, and machine learning. Bayesian inference needs the approximation of complicated integrals involving the posterior distribution. For this purpose, Monte Carlo (MC) methods such as Markov Chain Monte Carlo (MCMC) and Importance Sampling (IS) algorithms, are often employed. In this work, we introduce a compressed MC (C-MC) scheme in order to compress the information obtained previously by MC sampling. The basic C-MC version is based on the stratification technique, well-known for variance reduction purposes. Deterministic C-MC schemes are also presented, which provide very good performance. The compression problem is strictly related to moment matching approach applied in different filtering methods, often known as Gaussian quadrature rules or sigma-point methods. The connections to herding algorithms and quasi-Monte Carlo perspective are also discussed. C-MC is particularly useful in a distributed Bayesian inference framework, when cheap and fast communications with a central processor are required. Numerical results confirm the benefit of the introduced schemes, outperforming the corresponding benchmark methods.**

***Index Terms*—Bayesian Inference, Markov Chain Monte Carlo (MCMC), Importance Sampling, Particle Filtering, Gaussian Quadrature, Sigma Points, Herding Algorithms, quasi-Monte Carlo methods, Distributed Algorithms**

I. Introduction

An essential problem in signal processing, statistics, and machine learning is the estimation of unknown parameters or hidden functions from noisy observations. Within the Bayesian inference framework, these problems are addressed by constructing posterior probability density functions (pdfs) of the unknowns [2, 5, 44]. Unfortunately, the computation of statistical quantities related to these posterior distributions (such as moments or credible intervals) is analytically impossible in most real-world applications. As a consequence, developing approximate inference algorithms is of utmost interest. Monte Carlo (MC) sampling techniques are methodologies that come to the rescue for solving most difficult problems of inference [25, 43]. They are state-of-the-art tools for approximating complicated integrals involving sophisticated multidimensional target densities, based on random drawing of samples [43]. Markov Chain Monte Carlo (MCMC) algorithms, Importance Sampling (IS) schemes and its sequential version (particle filtering) are the most important classes of MC methods [25, 44].

E-mail: lmartino@ing.uc3m.es

Determinism and support points. In order to reduce the computational demand of the Monte Carlo methods and the variance of the corresponding estimators, deterministic procedures have been included within the sampling algorithms. In the so-called variance reduction techniques (e.g., conditioning, stratification, antithetic sampling, and control variates), negative correlation among the generated samples is forced/induced, hence obtaining more efficient estimators [40, 48]. In Quasi-Monte Carlo (QMC) methods, deterministic sequences of samples are employed based on the concept of *low-discrepancy*, avoiding all kind of randomness [12, 13, 39]. In the same line, deterministic approximations of the posterior distribution based on quadrature, cubature rules or unscented transformations are often applied, when are available [1, 20, 49, 44]. These techniques provide a set of particles deterministically chosen (often called *sigma points*), in order to match perfectly the estimation of a pre-established number of moments of the posterior density. Most of them are derived for the Gaussian distribution [44]. These techniques are usually used in filtering applications as extension of the standard Kalman filtering and as alternative to the particle filtering techniques based on MC sampling. The quadrature rules are very efficient since with N weighted particles summarized exactly the first $2N$ non-central moments. However, quadrature approximations are available only for certain target densities. Indeed, the true values of the moments must be known and a solution of an highly non-linear system must be provided. Clearly, this is possible only for specific target densities. More generally, the idea of sigma points is strictly connected to the need of *summarizing* a given distribution (and/or function) with a set of *representative, support points*, deterministically selected [29, 28]. This topic is an important task in computational statistics and has gained increasing attention in the last years: some relevant examples are the herding algorithms [9, 10, 22, 16], the studies about the representative points previously mentioned [28, 29], as well as the space-filling and experimental designs [15, 19, 41]. Some of them have been applied jointly with Monte Carlo schemes or used for numerical integration problems [22, 16].

Parallel and Distributed Computation. Distributed algorithms have become a very active topic during the past years favored by fast technological developments (e.g., see [8]). Considering L independent computing machines, we can distinguish two scenarios. In parallel computing, all the machines have access to the data, and algorithms are designed

to run fast on this set of L processors [3, 42, 35, 33]. In a complete distributed framework, each machine can process only a subset of data [38, 45]. The data splitting can be required due to the amount of data (Big Data) or for a demand of the specific application, as in a wireless sensor network with limited transmission power where local inference analyses are needed. In this scenario, specific techniques have been designed for providing a distributed or diffused inference depending if a central node is available or not, respectively [4, 23, 36, 37]. Generally, a common and key requirement is to properly summarize the local information before transmitting to other nodes [38, 3, 42].

Contribution. In this work, we introduce different schemes for compressing the information contained in N Monte Carlo samples into $M < N$ weighted particles, based on the stratification approach [40, 43]. In the Compressed Monte Carlo (C-MC) schemes, we replace the particle MC approximation obtained by N unweighted samples (e.g., generated by an MCMC algorithm) or weighted samples (e.g., generated by an IS algorithm), with another particle approximation with $M < N$ summary weighted samples. Clearly, we desire to reduce the loss of information in terms of moment matching, in the same fashion of the quadrature rules. In this sense, the M summary particles can be considered as approximate sigma points. Furthermore, for a specific choice of the partition (see the case of unweighted C-MC samples in Section III-F), an approximate low-discrepancy sequence (i.e., a QMC sequence) is obtained. Several alternatives are presented and discussed, including the random or deterministic selection of the summary particles.

Range of applicability. The Compressed Monte Carlo (C-MC) approach has a direct application in a parallel and/or distributed Bayesian framework, as graphically represented in Figure 1. Indeed, in this scenario, different local low-power nodes must transmit to a central node the results of their local Bayesian analysis, in order to provide a common complete inference [38, 3, 42]. The transmission should have the minimum possible cost and contain the maximum amount of information. Hence, the information must be properly compressed before be transmitted (see Section IV for further details). C-MC is an improvement of the bootstrap strategy, applied in different works regarding parallel sequential Monte Carlo schemes, where several resampled particles are transmitted jointly with the proper aggregated weight [3, 42, 47, 30]. Another possible application is inside the so-called parallel partitioned particle filters and multiple particle filters as alternative to the use of first moment estimators or the use of sigma points for approximating marginal posterior distributions [11, 34]. Furthermore, C-MC can be also applied within adaptive Monte Carlo schemes in order to obtain a good construction of the adaptive proposal density [7, 6, 32]. Indeed, C-MC can also return a mixture of densities as output, which can use as proposal pdf inside the adaptive MC technique [26, 6, 5]. Finally, note that sampling from the C-MC mixture can be employed as an alternative procedure to the resampling steps in particle filtering, in the same fashion of the works [21] and [24]. We provide some application examples in the numerical experiments.

Structure of the work. Section II introduces the basic setup of the Bayesian inference problem and describes the goal of the paper jointly with some possible solutions already presented in the literature. In Section III, we introduce the C-MC method whereas, in Section IV, we describe the application of C-MC in a distributed framework. Section V provides some numerical results, and some conclusions are contained in Section VI.

II. Background

A. Problem statement

In many real-world applications, the interest lies in obtaining information about the posterior probability density function (pdf) of set of unknown parameters given the observed data. Mathematically, denoting the vector of unknowns as $\mathbf{x} = [x_1, ..., x_{d_x}]^\top \in \mathcal{D} \subseteq \mathbb{R}^{d_X}$ and the observed data as $\mathbf{y} \in \mathbb{R}^{d_Y}$, the pdf is defined as

$$\bar{\pi}(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{Z(\mathbf{y})} \propto \pi(\mathbf{x}|\mathbf{y}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}), \quad (1)$$

where $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf, and $Z(\mathbf{y})$ is the normalization factor, that is usually called marginal likelihood or Bayesian evidence. From now on, we remove the dependence on $\mathbf{y}$ to simplify the notation. A particular integral involving the random variable $\mathbf{X} \sim \bar{\pi}(\mathbf{x})$ is then given by

$$I = E_\pi[h(\mathbf{X})] = \int_{\mathcal{D}} h(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x} = \frac{1}{Z}\int_{\mathcal{D}} h(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \quad (2)$$

where $h(\mathbf{x})$ can be any integrable function of $\mathbf{x}$.[1] For the sake of simplicity, we assume that the functions $h(\mathbf{x})$ and $\pi(\mathbf{x})$ are continuous in $\mathcal{D}$, and the integrand function, $h(\mathbf{x})\pi(\mathbf{x})$, in Eq. (2) is integrable. More generally, we are interested in finding a particle approximation $\widehat{\pi}^{(N)}(\mathbf{x})$ of the measure of $\bar{\pi}(\mathbf{x})$ [25]. In many practical scenarios, we cannot obtain an analytical solution of (2). One possible alternative is to use different deterministic quadrature rules or formulas based on sigma points for approximating the integral I [1, 20, 44]. However, these deterministic techniques are available only in specific scenarios, i.e., for some particular pdfs $\bar{\pi}(\mathbf{x})$. Hence, Monte Carlo schemes are often preferred and applied in order to estimate I and provide a particle approximation $\widehat{\pi}^{(N)}(\mathbf{x})$.

B. Monte Carlo sampling techniques

If it is possible to draw N independent samples, $\mathbf{x}_1, \dots, \mathbf{x}_N$, directly from $\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$ [43], then we can construct a particle approximation $\widehat{\pi}^{(N)}(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\delta(\mathbf{x} - \mathbf{x}_n)$ of the

[1] For the sake of simplicity, we have assumed $h(\mathbf{x}) : \mathbb{R}^{d_X} \to \mathbb{R}$ and the integral $I \in \mathbb{R}$ is a scalar value. However, a more proper assumption is $\mathbf{h}(\mathbf{x}) : \mathbb{R}^{d_X} \to \mathbb{R}^s$ and $\mathbf{I} \in \mathbb{R}^s$ where $s \geq 1$. All the techniques and results in this work are valid for the more general mapping with $s \geq 1$, but we keep the simpler notation for $s = 1$.

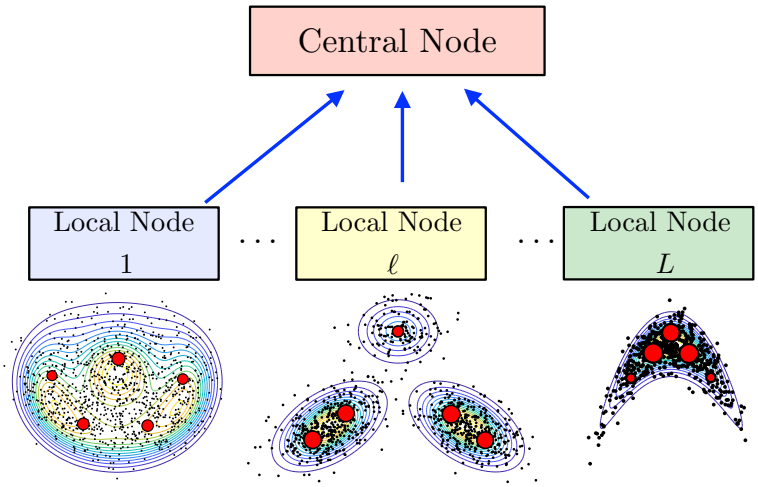


Figure 1. Graphical representation of a distributed Bayesian inference framework with L local computational nodes. Each local node addresses a posterior density, which is generally different in each node. If we consider just a parallel framework each node addresses the same posterior. In this example, a C-MC scheme compresses the information of N Monte Carlo samples (depicted by dots) with $M = 5$ summary weighted particles shown by circles (in each node). The size of the circles is proportional to the corresponding C-MC summary weight.

measure of $\bar{\pi}$. Therefore, replacing $\bar{\pi}(\mathbf{x})$ with $\widehat{\pi}^{(N)}(\mathbf{x})$ in Eq. (2), we obtain the standard Monte Carlo estimator of I, i.e,

$$\widehat{I}_N = \frac{1}{N}\sum_{n=1}^{N} h(\mathbf{x}_n). \tag{3}$$

If we are not able independent samples from $\bar{\pi}(\mathbf{x})$, alternatively we can use the MCMC algorithms [25, 43]. They generate correlated samples $\{\mathbf{x}_n\}_{n=1}^N$ that, after a burn-in period, are distributed according to $\bar{\pi}(\mathbf{x})$. Another possible approach is based on the importance sampling (IS) technique [43, 5]. Let us consider now N samples $\{\mathbf{x}_n\}_{n=1}^N$ drawn from a proposal pdf, $q(\mathbf{x})$, with heavier tails than the target, $\bar{\pi}(\mathbf{x})$. We assign a weight to each sample and then we can be normalized them as follows,

$$w_i = \frac{\pi(\mathbf{x}_i)}{q(\mathbf{x}_i)}, \qquad \bar{w}_i = \frac{w_i}{\sum_{j=1}^N w_j}, \tag{4}$$

with $i = 1, ..., N$. Therefore, the moment of interest can be approximated as

$$\widehat{I}_N = \frac{1}{N\widehat{Z}}\sum_{i=1}^{N} w_i h(\mathbf{x}_i) = \sum_{i=1}^{N} \bar{w}_i h(\mathbf{x}_i), \tag{5}$$

where $\widehat{Z} = \frac{1}{N}\sum_{j=1}^N w_j$ is a unbiased estimator of $Z = \int_{\mathcal{D}} \pi(\mathbf{x})d\mathbf{x}$ [43]. More generally, all the described Monte Carlo schemes give a particle approximation of the measure of $\bar{\pi}(\mathbf{x})$, i.e.,

$$\widehat{\pi}^{(N)}(\mathbf{x}) = \sum_{n=1}^{N} \bar{\beta}_n \delta(\mathbf{x} - \mathbf{x}_n), \text{ and } \widehat{I}_N = \sum_{n=1}^{N} \bar{\beta}_n h(\mathbf{x}_n), \tag{6}$$

where $\delta(\mathbf{x})$ is the Dirac delta function, $\bar{\beta}_n = \frac{1}{N}$ in the standard Monte Carlo and MCMC methods, and $\bar{\beta}_n = \bar{w}_n = \frac{1}{N\widehat{Z}}\frac{\pi(\mathbf{x}_n)}{q(\mathbf{x}_n)}$ in the IS technique. However, we always refer to the former case as unweighted Monte Carlo samples since the weights $\bar{\beta}_n = \frac{1}{N}$ are equal for all n. Tables I-II summarize the main notation of the work.[2]

C. Goal

In this work, we address the problem of summarizing the information contained in a set of N weighted or unweighted samples generated by a Monte Carlo sampling technique, with a smaller amount $M < N$ of weighted samples. This problem is strictly related to the more general challenge: summarizing the required information contained in a target density $\bar{\pi}(\mathbf{x})$, using a particle approximation (with the smallest amount of weighted particles). Clearly, in general, there is a loss of information. More precisely, given a Monte Carlo approximation $\widehat{\pi}^{(N)}(\mathbf{x})$ in Eq. (6), with N samples, we desire to construct another particle approximation

$$\widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^{M} \bar{a}_m \delta(\mathbf{x} - \mathbf{s}_m), \tag{7}$$

where $M < N$, $\sum_{m=1}^M \bar{a}_m = 1$, and $\mathbf{s}_m \in \mathcal{D}$, sharing with $\widehat{\pi}^{(N)}$ the required properties. The goal is to compress the statistical information contained in $\widehat{\pi}^{(N)}(\mathbf{x})$, reducing as much as possible the loss of information. We refer to $\bar{a}_m$ as summary weights and, to $\mathbf{s}_m$, as summary particles. The rate of compression is clearly given by $\eta = \frac{N}{M}$. A greater rate corresponds to a greater compression: when $\eta = 1$ we have no compression, when $\eta = N$ we have the maximum compression ($1 \leq \eta \leq N$).

D. Related works

In the literature, two families of possible solutions have been proposed for different and related purposes. The first one is based on a bootstrap technique, and can be always

[2]In this work, the words *sample* and *particle* are used as synonyms. Moreover, the expression "unweighted samples" is equivalent to "equally weighted samples" when referred to normalized weights. Note that the IS method provide also an estimator of the marginal likelihood, i.e., $\widehat{Z} = \frac{1}{N}\sum_{n=1}^N w_n$.

used. The second one is the moment-matching approach, which is available only in a limited amount of cases, i.e., the application is restricted only to some specific target pdfs $\bar{\pi}(\mathbf{x})$.

Bootstrap-based solution. Let us assume we have N unweighted samples. A simple approach consists in choosing uniformly M samples within the N possible ones. Similarly, in the case of weighted samples, this strategy consists in resampling M times within the set $\{\mathbf{x}_n\}_{n=1}^N$ according to the normalized importance weights $\bar{w}_n$, $n = 1, ..., N$ [3]. Then, a proper aggregated weight is associated to the resampled particles [3, 31, 30]. This kind of compression scheme has been widely used in different works (explicitly or implicitly), from distributed particle filtering methods and other sophisticated Monte Carlo algorithms [3, 42, 35, 47].

Moment-matching solution. For simplicity and without loss of generality, let us consider $d_X = 1$, i.e., $x \in \mathbb{R}$. For some specific types of target pdfs $\bar{\pi}(x)$ and specific domains $\mathcal{D}$, it is possible to obtain a deterministic particle approximation $\widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^M \rho_m \delta(x - s_m)$ where the weights ρ_m and the particles s_m are solutions of the nonlinear moment-matching system below,

$$\sum_{m=1}^M \rho_m s_m^r = \int_{\mathcal{D}} x^r \bar{\pi}(x) dx \quad \text{for} \quad r = 1, ..., R = 2M, \quad (8)$$

where the true values of the first $2M$ non-central moments, $\int_{\mathcal{D}} x^r \bar{\pi}(x) dx$, must be known. Hence we have $2M$ unknowns (the M weights ρ_m and the M particles s_m) and $R = 2M$ equations. Since the system is highly nonlinear, in general, the analytical solution is available only in few particular cases. These solutions are called *Gaussian Quadratures* [44], the corresponding deterministic particle approximation provide a perfect-matching with the first $2M$ moments (zero loss of information in the approximation of these moments). Quadrature rules and related sigma point methods have been widely applied within several generalized Kalman filtering techniques [1, 20, 44].

Compressed Monte Carlo (C-MC). In this work, we introduce a compression approach which improves the bootstrap strategy and extends the applicability of the moment-matching scheme, both described above. We consider the cases of compressing unweighted and weighted samples, e.g., the N samples have been previously generated by an MCMC algorithm or an IS technique, respectively. Figure 2 shows two examples of C-MC approximation with $M = 10$ summary particles. The size of the circles is proportional to the corresponding summary weight.

III. Compressed Monte Carlo (C-MC)

A. Stratification

The underlying grounds of C-MC are based on the so-called stratified sampling [27, 40]. The idea is to divide the support domain $\mathcal{D}$ of the random variable $\mathbf{X}$ into M separate and mutually exclusive regions. More specifically, let us consider an integer $M \in \mathbb{N}^+$, and a partition $\mathcal{P} = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_M\}$ of the state space with M disjoint subsets,

$$\begin{aligned} &\mathcal{X}_1 \cup \mathcal{X}_2 \cup ... \cup \mathcal{X}_M = \mathcal{D}, \\ &\mathcal{X}_i \cup \mathcal{X}_k = \emptyset \qquad i \neq k, \quad \forall i, j \in \{1, ..., M\}. \end{aligned} \quad (9)$$

We assume that all $\mathcal{X}_m$ are convex sets. Then, in the simplest version, one sample is drawn from each sub-region, and finally all the generated samples are combined for providing an estimator of $I = E_\pi[f(\mathbf{X})]$. We also denote the area of $\bar{\pi}(\mathbf{x})$ restricted in $\mathcal{X}_m$ as

$$\begin{aligned} \bar{a}_m = \mathbb{P}(\mathbf{X} \in \mathcal{X}_m) &= \int_{\mathcal{X}_m} \bar{\pi}(\mathbf{x}) d\mathbf{x} = \frac{1}{Z} \int_{\mathcal{X}_m} \pi(\mathbf{x}) d\mathbf{x}, \\ &= \frac{Z_m}{Z} = \frac{Z_m}{\sum_{j=1}^M Z_j}, \end{aligned} \quad (10)$$

where $Z_m = \int_{\mathcal{X}_m} \pi(\mathbf{x}) d\mathbf{x}$ and $Z = \sum_{j=1}^M Z_j = \int_{\mathcal{D}} \bar{\pi}(\mathbf{x}) d\mathbf{x}$. Clearly, note that $\sum_{m=1}^M \bar{a}_m = 1$. The target density can be expressed as a mixture of M non-overlapped densities,

$$\bar{\pi}(\mathbf{x}) = \sum_{m=1}^M \bar{a}_m \left[\frac{1}{\bar{a}_m} \bar{\pi}(\mathbf{x}) \mathbb{I}(\mathcal{X}_m) \right] = \sum_{m=1}^M \bar{a}_m \bar{\pi}_m(\mathbf{x}), \quad (11)$$

where

$$\bar{\pi}_m(\mathbf{x}) = \frac{1}{\bar{a}_m} \bar{\pi}(\mathbf{x}) \mathbb{I}(\mathcal{X}_m) = \frac{1}{Z_m} \pi(\mathbf{x}) \mathbb{I}(\mathcal{X}_m), \quad (12)$$

is a density, and $\mathbb{I}(\mathcal{X}_m)$ is an indicator variable that is 1 when $\mathbf{x} \in \mathcal{X}_m$ and 0 otherwise.

Stratified MC estimators. In order to simulate a sample $\mathbf{x}^*$ from $\bar{\pi}(\mathbf{x})$, we can draw an index $j^* \in \{1, ..., M\}$ according to the probability mass function $\bar{a}_m$, $m = 1, ..., M$ and the draw $\mathbf{x}^* \sim \bar{\pi}_{j^*}(\mathbf{x})$. Alternatively, we can yield an approximation of the measure of $\bar{\pi}$, drawing one sample from each region, i.e., $\mathbf{s}_m \sim \bar{\pi}_m(\mathbf{x})$, and then assign to each sample the weight $\bar{a}_m$, $m = 1, ..., M$. Therefore, in this scenario, the corresponding estimator of the integral I in Eq. (2) and the particle approximation are, respectively,

$$\widetilde{I}_M = \sum_{m=1}^M \bar{a}_m h(\mathbf{s}_m), \text{ and } \widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^M \bar{a}_m \delta(\mathbf{x} - \mathbf{s}_m) \quad (13)$$

where $\mathbf{s}_m \sim \bar{\pi}_m(\mathbf{x}) = \frac{1}{Z_m} \pi(\mathbf{x}) \mathbb{I}(\mathcal{X}_m)$, hence $\mathbf{s}_m \in \mathcal{X}_m$. See the `Supplementary Material`, for extensions and further details.

B. C-MC algorithms

Let us consider N unweighted samples $\mathcal{S}_{\texttt{tot}} = \{\mathbf{x}_n\}_{n=1}^N$ simulated from target pdf $\bar{\pi}(\mathbf{x})$, through some standard Monte Carlo scheme (either independent or correlated). Alternatively, if an importance sampling scheme has been applied, we have N weighted samples $\{\mathbf{x}_n, w_n\}_{n=1}^N$ (see Section II). Let $M < N$ be a constant value. Given the partition in Eq. (9), i.e., $\mathcal{X}_1 \cup \mathcal{X}_2 \cup ... \cup \mathcal{X}_M = \mathcal{D}$ formed by convex, disjoint sub-regions $\mathcal{X}_m$, we denote as

$$\mathcal{S}_m = \{\widetilde{\mathbf{x}}_{m,1}, ..., \widetilde{\mathbf{x}}_{m,J_m}\} \subseteq \mathcal{S}_{\texttt{tot}} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$$

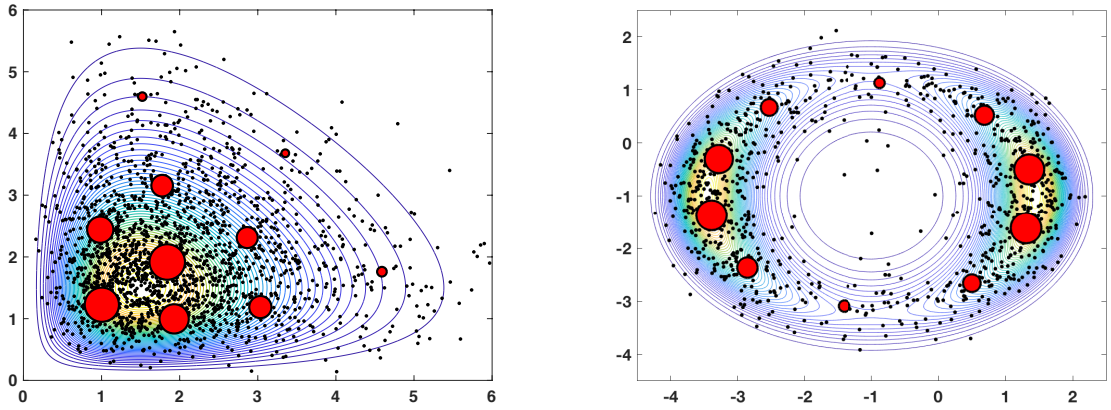


Figure 2. One run of a C-MC scheme with $M = 10$, for two different sets of $N = 10^3$ samples (represented by dots) distributed according to two different target $\bar{\pi}(\mathbf{x})$ (shown by the contour plots). The size of the circles is proportional to the corresponding summary weight.

the set of all the samples contained in the m-th region $\mathcal{X}_m$, i.e., $\widetilde{\mathbf{x}}_{m,j} \in \mathcal{X}_m$, with $m = 1, ..., M$, $j = 1, ...J_m$, where $J_m = |\mathcal{S}_m|$ is the cardinality of the m-th set $\mathcal{S}_m$. Clearly, we have $\sum_{m=1}^{M} J_m = N$, $\mathcal{S}_m \subset \mathcal{X}_m$, and $\cup_{m=1}^{M} \mathcal{S}_m = \mathcal{S}_{\texttt{tot}}$. We can compress the information contained in the particle approximation of Eq. (6), constructing an empirical stratified approximation based on M weighted particles $\{\mathbf{s}_m, \widehat{a}_m\}_{m=1}^{M}$, i.e.,

$$\widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^{M} \widehat{a}_m \delta(\mathbf{x} - \mathbf{s}_m), \tag{14}$$

so that for a specific moment the resulting estimator is

$$\widetilde{I}_M = \sum_{m=1}^{M} \widehat{a}_m h(\mathbf{s}_m). \tag{15}$$

Furthermore, in different applications, it is useful to define an aggregated weight W associated to the discrete measure $\widetilde{\pi}^{(M)}$, e.g., for further combination purposes in the distributed framework [3, 42, 30]. The definition of $\widehat{a}_m$ and $\mathbf{s}_m$ will be different, depending if we have unweighted or weighted Monte Carlo samples, as shown below:

Compressed Unweighted Sampling (CUS). In this scenario, each sample $\mathbf{s}_m \in \{\widetilde{\mathbf{x}}_{m,1}, ..., \widetilde{\mathbf{x}}_{m,J_m}\}$ is obtained resampling once within $\mathcal{S}_m$ with equal weights $\frac{1}{J_m}$. Namely, $\mathbf{s}_m$ is uniformly chosen in $\mathcal{S}_m$. Moreover, we define

$$\widehat{a}_m = \frac{J_m}{N}, \qquad W = N = \sum_{m=1}^{M} J_m, \tag{16}$$

that is an unbiased estimator of $\bar{a}_m$ in Eq. (10) since, by assumption, all the samples $\{\mathbf{x}_n\}_n^N$ are distributed according to $\bar{\pi}(\mathbf{x})$ and, hence, $\{\widetilde{\mathbf{x}}_{m,j}\}_{j=1}^{J_m}$ are distributed as $\bar{\pi}_m(\mathbf{x}) = \frac{1}{Z_m}\bar{\pi}(\mathbf{x})\mathbb{I}(\mathcal{X}_m)$, as well.

Compressed Importance Sampling (CIS). Let us denote the unnormalized and normalized weights of the samples of $\mathcal{S}_m$,

$$\widetilde{w}_{m,j} = \frac{\pi(\widetilde{\mathbf{x}}_{m,j})}{q(\widetilde{\mathbf{x}}_{m,j})}, \qquad \bar{w}_{m,j} = \frac{\widetilde{w}_{m,j}}{\sum_{k=1}^{J_m} \widetilde{w}_{m,k}}, \tag{17}$$

for $j = 1, ..., J_m$. Each sample $\mathbf{s}_m \in \{\widetilde{\mathbf{x}}_{m,1}, ..., \widetilde{\mathbf{x}}_{m,J_m}\}$ is obtained resampling once within $\mathcal{S}_m$ according to the probability mass function (pmf) defined by $\{\bar{w}_{m,j}\}_{j=1}^{J_m}$. Moreover, let us define

$$\widehat{Z}_m = \frac{1}{J_m} \sum_{j=1}^{J_m} \widetilde{w}_{m,j}, \tag{18}$$

that is an unbiased estimator of $\frac{1}{C_m}\int_{\mathcal{X}_m} \pi(\mathbf{x})d\mathbf{x}$ where $C_m = \int_{\mathcal{X}_m} q(\mathbf{x})d\mathbf{x}$. Then, we have

$$\widehat{a}_m = \frac{J_m \widehat{Z}_m}{N\widehat{Z}}, \qquad W = N\widehat{Z} = \sum_{m=1}^{M} J_m \widehat{Z}_m. \tag{19}$$

Note that the weights $\widehat{a}_m$ in both cases, for CUS and CIS, are unbiased estimators of $\bar{a}_m$ in Eq. (10). Let also define the unnormalized C-MC weights $a_m = \widehat{a}_m W$. Unlike in CUS, in CIS these weights, a_m, can be employed for estimating the marginal likelihood Z. Indeed, we can define

$$\begin{aligned} \widetilde{Z}_{\texttt{CIS}} &= \frac{1}{N}\sum_{m=1}^{M} a_m = \frac{1}{N}\sum_{m=1}^{M} J_m \widehat{Z}_m, \\ &= \frac{1}{N}\sum_{m=1}^{M} J_m \left[\frac{1}{J_m}\sum_{j=1}^{J_m} \widetilde{w}_{m,j}\right] = \widehat{Z}, \end{aligned} \tag{20}$$

recovering perfectly the IS estimator $\widehat{Z}$.[3]

Extensions and particular cases. Several summary particles can be also considered within a sub-region $\mathcal{X}_m$, instead of just one. Therefore, if we resample K_m particles uniformly in $\mathcal{X}_m$ for CUS or according to $\bar{w}_{m,j}$ for CIS, then the C-MC approximation will

[3]Note that, in the definition of the estimator $\widetilde{Z}_{\texttt{CIS}}$, it appears the factor $\frac{1}{N}$ instead of $\frac{1}{M}$.

be $\widetilde{\pi}^{(V)}(\mathbf{x}) = \sum_{m=1}^{M}\sum_{i=1}^{K_m}\frac{\widehat{a}_m}{K_m}\delta(\mathbf{x}-\mathbf{s}_{m,i})$, where $V = \sum_{m=1}^{M} K_m$. Note that CIS can be seen as an extension of Group Importance Sampling (GIS) [30] and the related approaches [3, 42, 47], where the different groups of samples belong to different sub-regions ($\mathcal{X}_m$) of the entire domain ($\mathcal{D}$). Namely, the bootstrap approach described in Section II-D can be seen as a special case of C-MC with a unique region $M = 1$, $\mathcal{X}_1 = \mathcal{D}$, and $V = K_1$ particles are resampled within $\{\mathbf{x}_n\}_{n=1}^{N}$. For the sake of simplicity, in the rest of the work we consider $K_m = 1$, for all m, and $V = M$, except when we explicitly state the opposite.

Proper partition and consistency. Let us focus in the way the partition is formed. A partition rule is proper if, when $M = N$, then $\mathcal{S}_m = \{\widetilde{\mathbf{x}}_m = \mathbf{x}_m\}$ (note that $m = n$ in this case), i.e., in the limit case of $M = N$ we consider all the MC samples as summary samples. Recall that, for $M < N$, the C-MC estimators are unbiased as shown in Eq. (**??**) (with $K_m = 1$ and $V = M$). Furthermore, if the partition rule is proper for $M = N$, C-MC estimators coincide with the classical Monte Carlo estimators. Hence, as $M \to N$ and $N \to \infty$, the consistency is ensured.

Save in transmission. Let us consider the parallel or distributed framework with a common central node. Note that, the unnormalized C-MC weights knowing all the $\{a_m\}_{m=1}^{M}$, we can recover $\widehat{a}_m = \frac{a_m}{W}$ with $W = \sum_{m=1}^{M} a_m$. In C-MC, only the M pairs $\{a_m, \mathbf{s}_m\}_{m=1}^{M}$ are transmitted to the central node, instead of the N pairs. Since, $\mathbf{x}, \mathbf{s} \in \mathbb{R}^{d_X}$, without compression, we need to transmit Nd_X scalar values in case on unweighted samples, or $N(d_X + 1)$ scalar values in the case of weighted samples. With the proposed compression, the transmission of only $M(d_X + 1)$ scalar values are required.

C. Loss

For the sake of simplicity and without loss of generality, in this section we consider the scalar case $d_X = 1$. Let us assume that we are particularly interested in the estimates of the first R moments of $\bar{\pi}$. The standard MC estimators, given the N samples, are denoted as

$$\widehat{I}_N^{(r)} = \sum_{n=1}^{N} \bar{\beta}_n x_n^r \approx \int_{\mathcal{D}} x^r \bar{\pi}(x)dx \quad \text{for} \quad r = 1, ..., R, \tag{21}$$

where the weights are $\bar{\beta}_n = \frac{1}{N}$ for the unweighted sample methods, and $\bar{\beta}_n = \bar{w}_n = \frac{1}{N\widehat{Z}}\frac{\pi(\mathbf{x}_n)}{q(\mathbf{x}_n)}$ for the IS technique. Thus, if we apply C-MC, we summarize the N samples with M pairs $\{s_m, \widehat{a}_m\}_{m=1}^{M}$ obtaining the estimators

$$\widetilde{I}_M^{(r)} = \sum_{m=1}^{M} \widehat{a}_m s_m^r. \tag{22}$$

We are interested in reproducing the values of $\widehat{I}_N^{(r)}$ using less samples with C-MC ($M < N$). Hence, for a specific r-th moment, the information loss for a C-MC scheme can be measured with the squared error, i.e., $\ell(r) = (\widehat{I}_N^{(r)} - \widetilde{I}_M^{(r)})^2$. More generally, considering $r = 1, ..., R$, we can define the loss as

$$\mathcal{L}_R = \sum_{r=1}^{R} \xi_r^2 \ell(r) = \sum_{r=1}^{R} \xi_r^2 \left(\widehat{I}_N^{(r)} - \widetilde{I}_M^{(r)}\right)^2, \tag{23}$$

which is a weighted average of the squared errors, with weights ξ_r^2. For instance, we can set $\xi_r^2 \propto \frac{1}{\left[\widehat{I}_N^{(r)}\right]^2}$ if $\widehat{I}_N^{(r)} \neq 0$ so that $\mathcal{L}_R$ is equivalent to a sum of the relative errors, or simply $\xi_r^2 = \frac{1}{R}$. Note that the loss depends on the chosen partition, as well as the specific realizations of $\{x_n\}_{n=1}^{N}$ and $\{s_m, \widehat{a}_m\}_{m=1}^{M}$.

CIS estimator of the marginal likelihood with zero-loss. In the weighted sample scenario, we have also the estimator of the marginal likelihood $\widehat{I}_N^{(0)} = \widehat{Z} = \frac{1}{N}\sum_{n=1}^{N} w_n$. The corresponding CIS estimator is $\widetilde{I}_M^{(0)} = \widetilde{Z}_{\texttt{CIS}} = \widehat{Z}$ shown in Eq. (20), so that

$$\ell(0) = \left(\widetilde{I}_M^{(0)} - \widehat{Z}\right)^2 = \left(\widetilde{Z}_{\texttt{CIS}} - \widehat{Z}\right)^2 = 0. \tag{24}$$

Namely, using CIS, we always recover the IS estimator of the marginal likelihood, without any loss.

Further considerations. We can express $\widehat{I}_N^{(r)}$ as a convex combination as partial MC estimators $\widehat{I}_{J_m}^{(r)} = \sum_{j=1}^{J_m} \bar{\gamma}_{m,j}\widetilde{x}_{m,j}^r$ considering only the samples in $\mathcal{X}_m$ where $\bar{\gamma}_{m,j} = \frac{1}{J_m}$ for CUS, whereas $\bar{\gamma}_{m,j} = \frac{1}{J_m \widehat{Z}_m}\frac{\pi(\widetilde{\mathbf{x}}_{m,j})}{q(\widetilde{\mathbf{x}}_{m,j})}$ for CIS. Namely, we can write (see Table II for recalling the notation)

$$\widehat{I}_N^{(r)} = \sum_{n=1}^{N} \bar{\beta}_n x_n^r = \sum_{m=1}^{M}\sum_{j=1}^{J_m} \bar{\beta}_{m,j}\widetilde{x}_{m,j}^r, \tag{25}$$

$$= \sum_{m=1}^{M} \widehat{a}_m \sum_{j=1}^{J_m} \bar{\gamma}_{m,j}\widetilde{x}_{m,j}^r = \sum_{m=1}^{M} \widehat{a}_m \widehat{I}_{J_m}^{(r)}, \tag{26}$$

where, in the first line, we have changed the representation from $\mathcal{S}_{\texttt{tot}} = \{x_n\}_{n=1}^{M} = \cup_{m=1}^{M}\mathcal{S}_m$ to the equivalent representation $\{\mathcal{S}_m\}_{m=1}^{M}$, with $\mathcal{S}_m = \{\widetilde{x}_{m,j}\}_{m=1}^{M}$. Note that, the equality $\bar{\beta}_{m,j} = \widehat{a}_m\bar{\gamma}_{m,j}$ holds (see Table II). Therefore, considering the loss at the r-th moment $\ell(r)$, we have

$$\ell(r) = \left(\widehat{I}_N^{(r)} - \widetilde{I}_M^{(r)}\right)^2 \tag{27}$$

$$= \left(\sum_{m=1}^{M} \widehat{a}_m \sum_{j=1}^{J_m} \bar{\gamma}_{m,j}\widetilde{x}_{m,j}^r - \sum_{m=1}^{M} \widehat{a}_m s_m^r\right)^2, \tag{28}$$

$$= \left(\sum_{m=1}^{M} \widehat{a}_m \left[\sum_{j=1}^{J_m} \bar{\gamma}_{m,j}\widetilde{x}_{m,j}^r - s_m^r\right]\right)^2, \tag{29}$$

$$= \left(\sum_{m=1}^{M} \widehat{a}_m e_m(r)\right)^2, \tag{30}$$

where $e_m(r) = \sum_{j=1}^{J_m} \bar{\gamma}_{m,j}\widetilde{x}_{m,j}^r - s_m^r$. This motivates an adaptive procedure for building a good partition, as shown in Section III-F.

D. Deterministic compression schemes

The CUS and CIS approaches described above choose randomly one summary particle in each sub-regions $\mathcal{X}_m$ (by a resampling step), based on the stratification idea. In the same fashion of the deterministic rules and sigma-point construction discussed in Section II-D, we can set

$$\mathbf{s}_m = \sum_{j=1}^{J_m} \bar{\gamma}_{m,j} \widetilde{\mathbf{x}}_{m,j} \approx E_\pi[\mathbf{X} \in \mathcal{X}_m], \tag{31}$$

where we recall $\bar{\gamma}_{m,j} = \frac{1}{J_m}$ for CUS, whereas $\bar{\gamma}_{m,j} = \frac{1}{J_m \widehat{Z}_m} \frac{\pi(\widetilde{\mathbf{x}}_{m,j})}{q(\widetilde{\mathbf{x}}_{m,j})}$ for CIS. Note that, with this choice of the summary particles, we have

$$\ell(1) = 0, \tag{32}$$

as shown in Appendix A. Moreover, with $\mathbf{s}_m$ in Eq. (31) and r is even, note that $e_m(r) = \text{var}_{\widehat{\pi}_m}\left[\mathbf{X}^{\frac{r}{2}}\right]$, i.e., the variance with respect to particle approximation $\widehat{\pi}_m(\mathbf{x}) = \sum_{j=1}^{J_m} \bar{\gamma}_{m,j} \delta(\mathbf{x} - \widetilde{\mathbf{x}}_{m,j})$, hence

$$\ell(r) = \left(\sum_{m=1}^{M} \widehat{a}_m \text{var}_{\widehat{\pi}_m}\left[\mathbf{X}^{\frac{r}{2}}\right] \right)^2, \qquad (r \text{ even}). \tag{33}$$

The choice in Eq. (31) is interesting since provides very good performance (see Section V) and also recall a deterministic quadrature rule with weighted nodes, that can be interpreted an approximate sigma-point construction [20, 44].

Zero-loss compression. Furthermore, if we are interested only in one specific integral I of the target pdf corresponding to the non-linear function $h(\mathbf{x})$, we can set

$$s_m = \sum_{j=1}^{J_m} \bar{\gamma}_{m,j} h(\widetilde{\mathbf{x}}_{m,j}) \approx E_\pi[h(\mathbf{X}) \in \mathcal{X}_m]. \tag{34}$$

Theorem 1. *Choosing s_m as in Eq.* (34) *(i.e., as ithe partial Monte Carlo estimator corresponding to the m-th sub-regions $\mathcal{X}_m$),*[4] *we have $\widetilde{I}_M = \widehat{I}_N$. Namely, the information loss with respect to the complete Monte Carlo estimator of specific integral involving $h(\mathbf{x})$ is zero.*

See Appendix A for the proof. Therefore, if we are interested in only one specific moment of $\bar{\pi}(\mathbf{x})$, we can obtain a perfect compression choosing the summary particles as in Eq. (34). Table II provides a list of the different weights used in this work.

E. Compression by kernel density estimation

In Eq. (14), we can replace the delta functions with Gaussian kernels $\mathcal{N}(\mathbf{x}|\mathbf{s}_m, \mathbf{\Sigma}_m)$, of mean $\mathbf{s}_m$ and with a $d_X \times d_X$ covariance matrix $\mathbf{\Sigma}_m$ the $d_X \times d_X$ obtained by an empirical estimation considering the samples in $\mathcal{X}_m$, i.e.,

$$\mathbf{\Sigma}_m = \sum_{j=1}^{J_m} \bar{\gamma}_{m,j} (\widetilde{\mathbf{x}}_{m,j} - \mathbf{s}_m)(\widetilde{\mathbf{x}}_{m,j} - \mathbf{s}_m)^\top, \tag{35}$$

[4]Note that in this case $s_m \in \mathbb{R}$ is a scalar value since, for simplicity, we have assumed $h(\mathbf{x}) : \mathbb{R}^{d_X} \to \mathbb{R}$, instead of the more general assumption $\mathbf{h}(\mathbf{x}) : \mathbb{R}^{d_X} \to \mathbb{R}^s$ with $s \geq 1$. All the considerations are also valid for $s \geq 1$.

where $\mathbf{s}_m$ is defined in Eq. (31). Hence, in this case, we have

$$\widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^{M} \widehat{a}_m \mathcal{N}(\mathbf{x}|\mathbf{s}_m, \mathbf{\Sigma}_m). \tag{36}$$

Recall the definition of $a_m = \widehat{a}_m W$ so that $\widehat{a}_m = \frac{a_m}{W}$ and $W = \sum_{m=1}^{M} a_m$. In this scenario, the M triplets $\{a_m, \mathbf{s}_m, \mathbf{\Sigma}_m\}_{m=1}^{M}$ must be transmitted in the central node. In this case, the transmission of $M(d_X^2 + d_X + 1)$ scalar values are required, i.e., Md_X^2 more values than with the C-MC particle approximation. However, if the deterministic schemes in Section III-D are employed, then we can have a better estimation of the second moments of the target pdf, that are underestimated with the deterministic C-MC approximations.

F. Choice of the partition

In this section, we discuss some examples of practical choices of the partition, and then a possible adaptive procedure. Given the N samples $\mathbf{x}_n = [x_{n,1}, ..., x_{n,d_X}]^\top \in \mathcal{D} \subseteq \mathbb{R}^{d_X}$, with $n = 1, ..., N$. Then, we list three practical choices from the simplest to the more sophisticated strategy:

- **P1** Random grid, where each component of the elements of the grid are contained within the intervals $\min_{n=1,...,N} x_{n,i}$ and $\max_{n=1,...,N} x_{n,i}$, for each $i = 1, ..., d_X$.
- **P2** Uniform deterministic grid, where each component of the elements of the grid are contained within the intervals $\min_{n=1,...,N} x_{n,i}$ and $\max_{n=1,...,N} x_{n,i}$, for each $i = 1, ..., d_X$.
- **P3** Voronoi partition obtained by a clustering algorithm with M clusters (e.g., the well-known k-means algorithm).

Adaptive procedure. Set $t = 0$ and choose an initial partition $\mathcal{P}_0 = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_{M_0}\}$ of the domain $\mathcal{D}$, with $M_0 = |\mathcal{P}_0|$ disjoint sub-regions, obtained with the approach **P2**, for instance. Decide also the stopping condition, choosing a maximum number of sub-regions $M_{\max} < N$ or a threshold for the loss L. Therefore, while $M_t \leq M_{\max}$ or $\mathcal{L}_R \geq L$ (where $\mathcal{L}_R$ is computed as in Eq. (23)), split the m^*-th sub-region, with

$$m^* = \arg\max_m \sum_{r=1}^{R} \xi_r \widehat{a}_m e_m(r). \tag{37}$$

Repeat the procedure above until achieve the desired stopping condition is reached. Recall that we define as *proper* any partition rule such that when $M = N$, then $\mathbf{s}_m = \widetilde{\mathbf{x}}_m = \mathbf{x}_n$ and $\widehat{a}_m = \bar{\beta}_n$ (note that $m = n$ in this case), i.e., in the limit case with $M = N$ we consider all the MC samples as summary samples.

Unweighted C-MC particles. Let us consider the CUS case, i.e., the compress of unweighted MC samples. If the partition is chosen such that $J_m = \frac{M}{N}$ for all $m = 1, ..., M$, then $\widehat{a}_m = \frac{1}{M}$. In this case, the partition is related to the empirical quantiles of the target distribution and we can interpret the C-MC particles as an approximate quasi-Monte Carlo (QMC) samples. Indeed, as the number of MC samples N grows, the distribution of the nodes $\mathbf{s}_m$ follows the definition of low-discrepancy [39]. Furthermore, since

$\widehat{a}_m = \frac{1}{M}$ for all m then, in a distributed scenario, the transmission of summary weights can be avoided: the only information still required is the aggregated weight $W = N$, as we show in the next section. However, we recall that the performance in terms of information loss (see Section III-C) depends also to the empirical variance $\widehat{\sigma}_m^2$, or more generally $e_m(r)$, of each sub-region.

IV. Application to distributed inference

In this section, we consider L independent computational nodes where the Monte Carlo computation is performed in parallel. Moreover, we consider a central node where the transmitted local information is properly combined, as represented in Figure 1. We distinguish three different scenarios. In the first one, from now on referred as parallel framework, the same dataset $\mathbf{y} \in \mathbb{R}^{d_Y}$ and the same model is shared by all the local nodes. Thus, all the L nodes address the same inference problem, i.e., they deal with the same posterior density. In the second scenario, referred as model selection case, all the nodes have accessed to the entire dataset $\mathbf{y}$, but each local node considers a different possible model (different likelihood and/or prior functions), hence they deal with different posteriors. The third case is the distributed scenario, where the observed data are divided over the L local nodes, $\mathbf{y} = [\mathbf{y}_1, ...\mathbf{y}_L]^\top$. Hence, each node addresses a different sub-posterior density which considers only a subset of the data, $\mathbf{y}_\ell \in \mathbb{R}^{d_\ell}$ (note that $\sum_{\ell=1}^L d_\ell = d_Y$). In these frameworks, a particle compression is often required for reducing the computational and the transmission cost. Below, we provide more details.

A. Parallel framework

We assume the use of N_ℓ particles $\mathbf{x}_1^{(\ell)}, ..., \mathbf{x}_{N_\ell}^{(\ell)}$ in each local node. First of all, we consider the transmission of all the particles of the central node, without any compression. In this case, the complete Monte Carlo approximation with $\sum_{\ell=1}^L N_\ell$ particles can be expressed as

$$\widehat{\pi}_{\mathtt{tot}}(\mathbf{x}) = \sum_{\ell=1}^{L} \frac{W_\ell}{\sum_{j=1}^{N_\ell} W_j} \sum_{n=1}^{N_\ell} \bar{\beta}_n^{(\ell)} \delta(\mathbf{x} - \mathbf{x}_n^{(\ell)}) \quad (38)$$

$$= \sum_{\ell=1}^{L} \bar{\rho}_\ell \; \widehat{\pi}_\ell^{(N_\ell)}(\mathbf{x}), \quad (39)$$

where $\bar{\rho}_\ell = \frac{W_\ell}{\sum_{j=1}^{N_\ell} W_j}$, and $\bar{\beta}_n^{(\ell)} = \frac{1}{N_\ell}$, $W_\ell = N_\ell$ in the case of unweighted samples, or $\bar{\beta}_n^{(\ell)} = \frac{1}{N_\ell \widehat{Z}^{(\ell)}} \frac{\pi(\mathbf{x}_n^{(\ell)})}{q(\mathbf{x}_n^{(\ell)})}$, $W_\ell = N_\ell \widehat{Z}^{(\ell)}$ in the case of weighted samples. Therefore, the complete Monte Carlo approximation $\widehat{\pi}_{\mathtt{tot}}^{(LN)}(\mathbf{x})$ is a convex combination of the L local particle approximations $\widehat{\pi}_\ell^{(N)}(\mathbf{x})$. If we apply a compression scheme transmitting $M_\ell < N_\ell$ samples, $\widetilde{\pi}_\ell^{(M_\ell)}(\mathbf{x})$ as in Eq. (14) or (36), then the joint particle approximation in the central node is

$$\widetilde{\pi}_{\mathtt{tot}}(\mathbf{x}) = \sum_{\ell=1}^{L} \bar{\rho}_\ell \; \widetilde{\pi}_\ell^{(M_\ell)}(\mathbf{x}). \quad (40)$$

We aim to have a small loss of information between the particle approximations, $\widetilde{\pi}_{\mathtt{tot}}(\mathbf{x})$ and $\widehat{\pi}_{\mathtt{tot}}(\mathbf{x})$. In [3, 42, 47, 30], the bootstrap strategy described in Section II-D is applied for the compression. In the numerical experiments, we compare the performance of this strategy with the C-MC approach.

B. Model Selection

The model selection application is an easy extension of the parallel framework. In this scenario, all the nodes process the entire set of data $\mathbf{y}$, but each local node considers a different possible model $\mathcal{M}_\ell$, hence they address different posterior distributions $\bar{\pi}(\mathbf{x}|\mathbf{y}, \mathcal{M}_\ell)$. In order to tackle this problem, based on the Bayesian Model Averaging (BMA) approach, we need an estimation of the marginal likelihood of each model $\widehat{Z}^{(\ell)}$ (e.g., see [46, 33]). Therefore, for this reason, it is preferable to apply an IS scheme where an estimator of the marginal likelihood is easily provided. The equations are the same of the previous parallel scenario, i.e., we have $\widehat{\pi}_{\mathtt{tot}}(\mathbf{x}) = \sum_{\ell=1}^L \frac{N_\ell \widehat{Z}^{(\ell)}}{\sum_{k=1}^L N_k \widehat{Z}^{(k)}} \; \widehat{\pi}_\ell^{(N_\ell)}(\mathbf{x})$ without compression, and $\widetilde{\pi}_{\mathtt{tot}}(\mathbf{x}) = \sum_{\ell=1}^L \frac{N_\ell \widehat{Z}^{(\ell)}}{\sum_{k=1}^L N_k \widehat{Z}^{(k)}} \; \widetilde{\pi}_\ell^{(M_\ell)}(\mathbf{x})$, with compression. In this scenario, $\bar{\rho}_\ell = \frac{N_\ell \widehat{Z}^{(\ell)}}{\sum_{k=1}^L N_k \widehat{Z}^{(k)}}$, for $\ell = 1, ..., L$, represents an approximation of the posterior probability mass function (pmf) of the model given the data, i.e., $p(\mathcal{M}_\ell|\mathbf{y})$.

C. Distributed framework

For the sake of simplicity, let us assume $N_\ell = N$ and $M_\ell = M$, for all $\ell = 1, ..., L$. In this case, all the nodes consider the same model as in the parallel scenario, but each local node can process only a portion of the observed data, $\mathbf{y}_\ell \in \mathbb{R}^{d_\ell}$, with $\sum_{\ell=1}^L d_\ell = d_Y$. Considering a disjoint subsets of data and a split contribution of the prior as in [38], the complete posterior can be factorized as

$$\bar{\pi}_{\mathtt{tot}}(\mathbf{x}) \propto \prod_{\ell=1}^{L} \bar{\pi}_\ell(\mathbf{x}) \quad (41)$$

In different works [38, 45], local approximations of the sub-posteriors are provided and transmitted to the central node, obtaining

$$\widehat{\pi}_{\mathtt{tot}}(\mathbf{x}) \propto \prod_{\ell=1}^{L} \widehat{\pi}_\ell^{(N)}(\mathbf{x}). \quad (42)$$

The simplest approach considers Gaussian local approximations [38, 45]. A more sophisticated approach proposed in [38, Section 3.2] considers a mixture of Gaussian pdfs as KDE local approximation using all the N samples in the node, i.e.,

$$\widehat{\pi}_\ell^{(N)}(\mathbf{x}) = \sum_{n=1}^{N} \bar{\beta}_n^{(\ell)} \mathcal{N}(\mathbf{x}|\mathbf{x}_n^{(\ell)}, \delta\mathbf{I}), \quad (43)$$

with $\delta > 0$ and $\mathbf{I}$ is a $d_X \times d_X$ identity matrix. It is straightforward to see that $\widehat{\pi}_{\mathtt{tot}}(\mathbf{x})$ can be expressed as a mixture of N^L Gaussian components [38, 18]. It is possible to draw from this mixture, but clearly the cost depends of the number of N^L components [18]. Therefore, here the

advantage of using a compressed local mixture, $\widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^{M} \widehat{a}_m \mathcal{N}(\mathbf{x}|\mathbf{s}_m, \mathbf{\Sigma}_m)$ with $M < N$, is even more apparent than in the parallel scenarios described above. Indeed, using C-MC, we obtain $\widehat{\pi}_{\texttt{tot}}(\mathbf{x}) \propto \prod_{\ell=1}^{L} \widetilde{\pi}_{\ell}^{(M)}(\mathbf{x})$, that can be expressed as a mixture of M^L Gaussian pdfs [38, 18].

V. Numerical experiments

In the section, we test the proposed C-MC techniques and compare the performance with the corresponding benchmark scheme, i.e., the bootstrap solution proposed in [3] and also used in [42, 35, 47, 30]. In the first experiment, we apply the compression techniques to two sets of Monte Carlo samples. In the second experiment, we consider a localization problem in a wireless sensor network and the use of L local processors.

A. First numerical analysis

Let us consider for simplicity $x \in \mathbb{R}$, i.e., $d_X = 1$. Moreover, we consider two possible target densities: the first one is a Gamma pdf

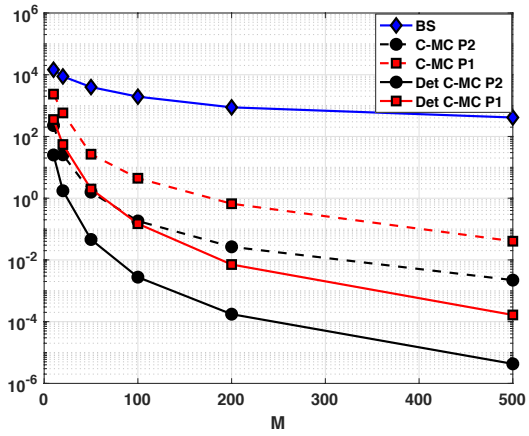
$$\bar{\pi}(x) \propto x^{\alpha-1} \exp\left(-\frac{x}{\kappa}\right), \tag{44}$$

with $\alpha = 4$ and $\kappa = 0.5$, and the second one is a mixture of two Gaussians,

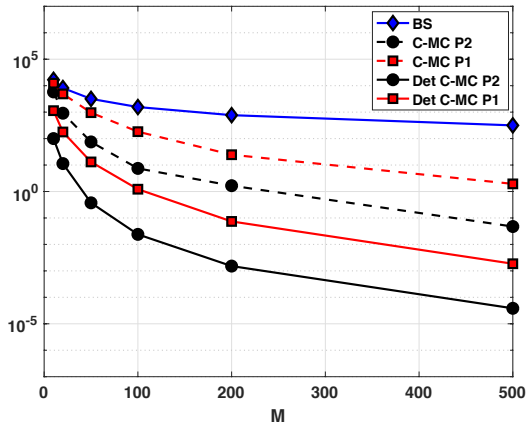
$$\bar{\pi}(x) = \frac{1}{2}\mathcal{N}(x|-2,1) + \frac{1}{2}\mathcal{N}(x|4,0.25). \tag{45}$$

Experiments. We generate $N = 10^5$ Monte Carlo samples from both and compare the bootstrap strategy (BS) with different C-MC schemes. More specifically, we consider two kind of partition procedures: random (P1) and uniform (P2) described in Section III-F. Furthermore we compare the stochastic and the deterministic choices of the summary particles $\mathbf{s}_m$ described in Section III. Therefore, for the deterministic C-MC we refer to the use of Eq. (31) for $\mathbf{s}_m$. We repeat the experiment 500 independent runs and average the results. At each run, we compute the loss $\mathcal{L}_5$ with $\xi_r^2 = 1$, for $r = 1, ..., 5$ (i.e., the loss in the first 5 moments) provided by the different techniques. Figure 3 depicts the averaged $\mathcal{L}_5$ as function of the number M of summary particles. Figure 3-**(a)** refers to the Gamma target pdf, whereas Figure 3-**(b)** corresponds to the Gaussian mixture pdf. The results of the BS method are displayed with triangles. The stochastic C-MC schemes are shown with dashed lines, whereas the deterministic C-MC schemes with solid lines.

Discussion. In all cases, C-MC outperforms BS and the deterministic C-MC schemes provide the better results. Clearly, the partition P2 (depicted with circles) outperforms P1 (shown with squares). Note that P1 represents the simplest and perhaps the worst possible construction of the partition. However, it is important to remark that the C-MC schemes, even with P1, outperform the BS method.



(a) Gamma target pdf



(b) Mixture target pdf

Figure 3. The loss $\mathcal{L}_5$ as function of M. The results obtained by the bootstrap strategy [3, 42, 30] in Section II-D is depicted with a solid line and rhombuses. The results of C-MC with a random partition (P1) and with a grid partition (P2) are shown by squares and circles, respectively. The results obtained with the deterministic choice of $\mathbf{s}_m$ in Eq. (31) are shown with solid lines (squares and circles), whereas the results random choice of $\mathbf{s}_m$ are provided with dashed lines (squares and circles).

B. Localization in a sensor network with Parallel AIS schemes

In this section, we test the C-MC technique considering the problem of positioning a target in $\mathbb{R}^2$ ($d_X = 2$) using a range measurements in a wireless sensor network [14, 17]. Specifically, the target position is modeled as a random vector $\mathbf{X} = [X_1, X_2]^\top$, hence the actual position of the target is a specific realization $\mathbf{X} = \mathbf{x}$. The data (range measurements) are obtained from 3 sensors located at $\mathbf{h}_1 = [3, -8]^\top$, $\mathbf{h}_2 = [10, 0]^\top$, $\mathbf{h}_3 = [0, 10]^\top$, as shown in Figure 4-**(d)**. The likelihood function is induced by the following observation model,

$$Y_j = 20 \log\left(||\mathbf{x} - \mathbf{h}_j||\right) + B_j, \quad j = 1, 2, 3, \tag{46}$$

where $B_j \sim \mathcal{N}(b_j; 0, \lambda_j^2)$. We consider the true position of the target as $\mathbf{x}^* = [x_1^* = 2.5, x_2^* = 2.5]^\top$ and set $\lambda_j = 6$. Then, we generate one measurement y_j from each sensor according to the model in Eq. (46), obtaining the vector $\mathbf{y} = [y_1, y_2, y_3]$. Assuming a uniform prior in the rectangle $\mathcal{R}_z = [-30, 30]^2$, then the posterior density is

$$\bar{\pi}(\mathbf{x}) \propto \left[\prod_{j=1}^{3} \exp\left(-\frac{1}{2\lambda_j^2}(y_j - 20\log\left(||\mathbf{z} - \mathbf{h}_j||\right)^2\right)\right] \mathbb{I}_{\mathcal{R}_z}(\mathbf{x}), \tag{47}$$

where $\mathbb{I}_{\mathcal{R}_z}(\mathbf{x})$ is an indicator function that is 1 if $\mathbf{x} \in \mathcal{R}_z$, otherwise is 0.

Parallel setup. We assume L local computational nodes. At each one, we run an adaptive importance sampler, specifically a standard Population Monte Carlo (PMC) scheme [7]. Each PMC delivers N weighted samples as an approximation of the posterior of Eq. (47), after a certain number of iterations [5]. Therefore, we have $\widehat{\pi}_\ell^{(N)}$ local approximations of N particles. In this setting, we have a clear improvement in term of computational times since L different PMC algorithms are run in parallel. When all the samples are transmitted to the central node, we obtain a complete particle approximation $\widehat{\pi}_{\texttt{tot}}^{(NL)}$ as in Eq. (38). However, in general due to the transmission cost, a particle compression is applied. In this case, we have L local approximations $\widetilde{\pi}_\ell^{(M)}$, and the central node performs the fusion obtaining $\widetilde{\pi}_{\texttt{tot}}^{(ML)}$ as in Eq. (40). We measure the quality of the approximation $\widetilde{\pi}_{\texttt{tot}}^{(ML)}$ computing the loss (i.e., mean square error) in the estimation of the mean vector, the covariance matrix, skewness, and kurtosis vectors (i.e., overall 9 scalar values) with respect to $\widehat{\pi}_{\texttt{tot}}^{(NL)}$. We compare the bootstrap strategy (BS) suggested in [3, 42, 47, 30] and C-MC. For building the partition for C-MC, in each local node we apply N resampling steps and perform a k-means clustering with M clusters. Thus, the partition is given by the M Voronoi regions. Then, we consider again the weighted samples produced by the PMC and build the summary weights $\widehat{a}_m$ and summary samples $\mathbf{s}_m$ for each Voronoi region. We average the results over 200 independent runs.

Experiments. We set $L = 10$. The losses of BS (triangles) and C-MC (circles) for different values of M and N are depicted in Figures 4 **(a)**-**(b)**-**(c)**. More specifically, in Figure 4-**(a)** we set $N = 1000$ and vary M. In Figure 4-**(b)**, we vary M keeping fixed the compression rate $\eta = \frac{N}{M} = 100$, i.e., when M grows also N is increased. Finally, in Figure 4-**(c)**, we set $M = 10$, and vary N.

Discussion. First of all, we can observe that C-MC always outperforms BS providing the small loss in any scenario. The increase of M has always a positive impact as shown in Figures 4-**(a)**-**(b)**. In Figure 4-**(c)**, the compression rate $\eta = \frac{N}{M}$ is increasing since M is fixed and N grows, so that we expect that the performance should become worse as N grows. However, in a first moment, the increase of N helps both schemes, C-MC and BS, since a better partition can be build with a greater N in C-MC by clustering, and the resampling steps used in bootstrap improves its performance with a greater N in BS. Moreover, in this scenario, the increase of N seems to have more positive impact in the BS technique. However, Figure 4-**(b)** shows that, if the compression rate $\eta = \frac{N}{M}$ is maintained fixed, then C-MC obtains a better improvement. Figure 4-**(d)** depicts the wireless sensor network and the contour plot of the posterior pdf. Additionally, one run of C-MC with $M = 15$ is shown with circles. The size of the circles is proportional to the corresponding summary weight.

VI. Conclusions

In this work, we have introduced a novel efficient scheme for summarizing the information provided by Monte Carlo sampling algorithms. This problem is strictly related to moment matching approach used in different filtering methods and applicable only for certain target densities. The proposed technique is particularly useful in a distributed Bayesian inference framework. Different variants have been derived and discussed.

The C-MC proposed schemes have been tested in two numerical experiments. In the first example, we have considered two different target densities. In the second numerical analysis, we have considered a localization problem in a wireless sensor network. The inference is performed by using a distributed framework with L local processors. The results have shown that C-MC techniques always outperform the corresponding benchmark method. The deterministic C-MC scheme appears particularly efficient. Note that it can be also interpreted as an approximate sigma-point approach. Furthermore, the proposed C-MC methodology could be applied within advanced filtering schemes, for designing an efficient proposal density and/or as an alternative resampling procedure.

References

[1] I. Arasaratnam and S. Haykin. Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 54(6):1254–1269, 2009.

[2] M. S. Arulumpalam, S. Maskell, N. Gordon, and T. Klapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions Signal Processing*, 50(2):174–188, February 2002.

[3] M. Bolić, P. M. Djurić, and S. Hong. Resampling algorithms and architectures for distributed particle filters. *IEEE Transactions Signal Processing*, 53(7):2442–2450, 2005.

[4] C. J. Bordin and M. G. S. Bruno. Consensus-based distributed particle filtering algorithms for cooperative blind equalization in receiver networks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3968–3971, 2011.

[5] 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.

[6] 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.

[7] 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.

[8] M. Cetin, L. Chen, J. W. Fisher III, A. T. Ihler, R. L. Moses, M. J. Wainwright, and A. S. Willsky. Distributed fusion in sensor networks. *IEEE Signal Processing Magazine*, 23(4):56–69, July 2006.

[9] W. Ye Chen, L. Mackey, J. Gorham, F. X. Briol, and C. J. Oates. Stein Points. *arXiv:1803.10161*, pages 1–31, 2018.

[10] 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.

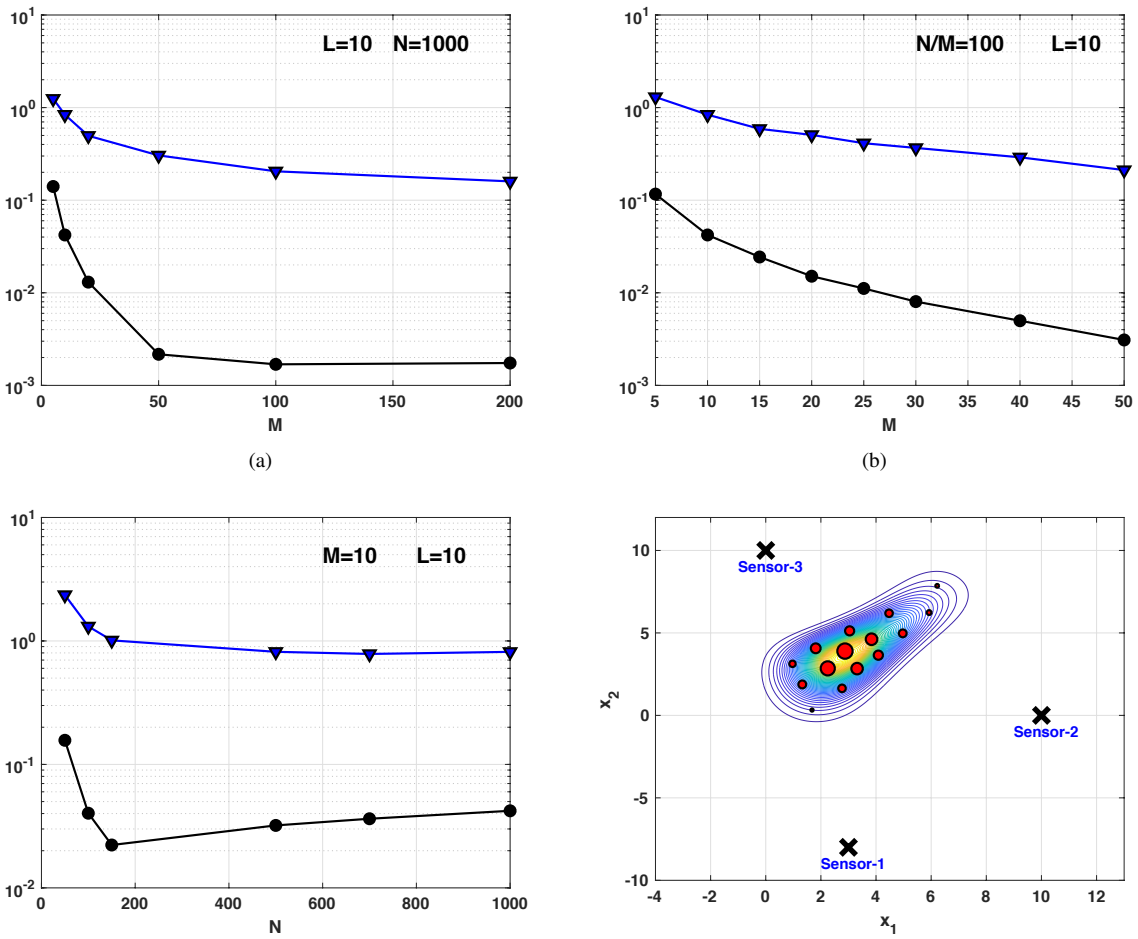


Figure 4. **(a)-(b)-(c)** Results in terms of information loss for the localization problem in wireless sensor network: C-MC is shown with circles and the bootstrap strategy with triangles. **(d)** The wireless sensor network and the contour plot of the posterior target pdf. One run of C-MC is shown with circles, $M = 15$. The size of the circles is proportional to the corresponding summary weight.

[11] P. M. Djuric, T. Lu, and M. F. Bugallo. Multiple particle filtering. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, pages 1181–1184, 2007.

[12] P. Fearnhead. Using random Quasi-Monte Carlo within particle filters, with application to financial time series. *Journal of Computational and Graphical Statistics*, 14(4):751–769, 2005.

[13] M. Gerber and N. Chopin. Sequential quasi Monte Carlo. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(3):509–579, 2015.

[14] R. P. Guan, B. Ristic, L. Wang, and R. Evans. Monte Carlo localisation of a mobile robot using a Doppler-Azimuth radar. *Automatica*, 97:161 – 166, 2018.

[15] P. Hennig and R. Garnett. Exact sampling from determinantal point processes. *arXiv:1609.06840*, pages 1–9, 2016.

[16] F. Huszár and D. Duvenaud. Optimally-weighted herding is Bayesian quadrature. *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence (UAI-12)*, pages 377–386, 2012.

[17] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky. Nonparametric belief propagation for self-localization of sensor networks. *IEEE Transactions on Selected Areas in Communications*, 23(4):809–819, April 2005.

[18] A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky. Efficient multiscale sampling from products of Gaussian Mixtures. *Advances in Neural Information Processing Systems (NIPS)*, pages 1–8, 2004.

[19] M. E. Johnson, L. M. Moore, and D. Ylvisaker. Minimax and maximin distance design. *Journal of Statistical Planning and Inference*, 26(2):131–148, 1990.

[20] S. J. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(2):401–422, March 2004.

[21] J. Kotecha and Petar M. Djurić. Gaussian sum particle filtering. *IEEE Transactions Signal Processing*, 51(10):2602–2612, October 2003.

[22] S. Lacoste-Julien, F. Lindsten, and F. Bach. Sequential kernel herding: Frank-Wolfe optimization for particle filtering. *In Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, page 544552, 2015.

[23] S. H. Lee and M. West. Convergence of the markov chain distributed particle filter (mcdpf). *IEEE Transactions on*

Signal Processing, 61(4):801–812, 2013.

[24] T. Li, T. P. Sattar, and S. Sun. Deterministic resampling: Unbiased sampling to avoid sample impoverishment in particle filters. *Signal Processing*, 92(7):1637–1645, 2012.

[25] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2004.

[26] D. Luengo and L. Martino. Fully adaptive Gaussian mixture Metropolis-Hastings algorithm. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[27] P. LEcuyer. Efficiency improvement and variance reduction. *In Proceedings of the 1994 Winter Simulation Conference*, pages 122–132, 1994.

[28] S. Mak and V. R. Joseph. Projected support points: a new method for high-dimensional data reduction. *arXiv:1708.06897*, pages 1–48, 2018.

[29] S. Mak and V. R. Joseph. Support points. *(to appear) Annals of Statistics, arXiv:1609.01811*, pages 1–55, 2018.

[30] L. Martino, V. Elvira, and G. Camps-Valls. Group Importance Sampling for Particle Filtering and MCMC. *Digital Signal Processing*, 82:133–151, 2018.

[31] 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.

[32] 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.

[33] 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.

[34] L. A. Úbeda Medina. Robust techniques for multiple target tracking and fully adaptive radar. *Phd Thesis, Universidad Politecnica de Madrid (UPM)*, pages 1 – 254, 2018.

[35] J. Míguez and M. A. Vázquez. A proof of uniform convergence over time for a distributed particle filter. *Signal Processing*, 122:152–163, 2016.

[36] A. Mohammadi and A. Asif. Distributed particle filter implementation with intermittent/irregular consensus convergence. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 61:2572–2587, 2013.

[37] A. Mohammadi and A. Asif. Diffusive particle filtering for distributed multisensor estimation. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3801–3805, 2016.

[38] W. Neiswanger, C. Wang, and E. Xing. Asymptotically exact, embarrassingly parallel MCMC. *arXiv:1311.4780*, 2013.

[39] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial Mathematics, 1992.

[40] A. Owen. *Monte Carlo theory, methods and examples*. http://statweb.stanford.edu/∼owen/mc/, 2013.

[41] Luc Pronzato. Minimax and maximin space-filling designs: some properties and methods for construction. *Journal de la Societe Franaise de Statistique, Societe Franaise de Statistique et Societe Mathematique de France*, 158(1):7–36, 2017.

[42] J. Read, K. Achutegui, and J. Míguez. A distributed particle filter for nonlinear tracking in wireless sensor networks. *Signal Processing*, 98:121 – 134, 2014.

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

[44] S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[45] 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. *International Journal of Management Science and Engineering Management*, 11(2):78–88, 2016.

[46] 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.

[47] C. Verg, C. Dubarry, P. Del Moral, and E. Moulines. On parallel implementation of sequential Monte Carlo methods: the island particle model. *Statistics and Computing*, 25(2):243–260, 2015.

[48] J. R. Wilson. Variance reduction techniques for digital simulation. *American Journal of Mathematical and Management Sciences*, 4(3):277–312, 1984.

[49] Y. Wu, D. Hu, M. Wu, and X. Hu. A numerical-integration perspective on Gaussian filters. *IEEE Transactions on Signal Processing*, 54(8):2910–2921, 2006.

APPENDIX A
ZERO-LOSS COMPRESSION FOR A SPECIFIC INTEGRAL I

We have stated that with the choice $s_m = \sum_{j=1}^{J_m} \bar{\gamma}_{m,j} h(\widetilde{\mathbf{x}}_{m,j})$ in (34) with a generic nonlinear function $h(\mathbf{x})$, then we have $\widetilde{I}_M \equiv \widehat{I}_N$. It is straightforward to see it, for the special case in Eq. (31), when $h(\mathbf{x})$ is a linear function. In the case of CUS, we have

$$\widetilde{I}_M = \sum_{m=1}^{M} \widehat{a}_m s_m = \sum_{m=1}^{M} \frac{J_m}{N} s_m \quad (48)$$

$$= \frac{1}{N} \sum_{m=1}^{M} J_m \left[\sum_{j=1}^{J_m} \bar{\gamma}_{m,j} h(\widetilde{\mathbf{x}}_{m,j}) \right], \quad (49)$$

$$= \frac{1}{N} \sum_{m=1}^{M} J_m \left[\sum_{j=1}^{J_m} \frac{1}{J_m} h(\widetilde{\mathbf{x}}_{m,j}) \right] = \widehat{I}_N. \quad (50)$$

Whereas, for CIS, we have

$$\widetilde{I}_M = \sum_{m=1}^{M} \widehat{a}_m s_m = \sum_{m=1}^{M} \frac{J_m \widehat{Z}_m}{N\widehat{Z}} s_m \tag{51}$$

$$= \sum_{m=1}^{M} \frac{J_m \widehat{Z}_m}{N\widehat{Z}} \left[\sum_{j=1}^{J_m} \bar{\gamma}_{m,j} h(\widetilde{\mathbf{x}}_{m,j}) \right], \tag{52}$$

$$= \sum_{m=1}^{M} \frac{J_m \widehat{Z}_m}{N\widehat{Z}} \left[\sum_{j=1}^{J_m} \frac{1}{J_m \widehat{Z}_m} \frac{\pi(\widetilde{\mathbf{x}}_{m,j})}{q(\widetilde{\mathbf{x}}_{m,j})} h(\widetilde{\mathbf{x}}_{m,j}) \right],$$

$$= \frac{1}{N\widehat{Z}} \sum_{m=1}^{M} \sum_{j=1}^{J_m} \frac{\pi(\widetilde{\mathbf{x}}_{m,j})}{q(\widetilde{\mathbf{x}}_{m,j})} h(\widetilde{\mathbf{x}}_{m,j}), \tag{53}$$

$$= \frac{1}{N\widehat{Z}} \sum_{n=1}^{N} \frac{\pi(\mathbf{x}_n)}{q(\mathbf{x}_n)} h(\mathbf{x}_n) = \widehat{I}_N. \tag{54}$$

Table I
MAIN NOTATION OF THE WORK.

$\mathbf{x} = [x_1, \dots, x_{d_X}]$	Variable of interest, $\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^{d_X}$
$\mathcal{X}_m$	sub-region of the domain, $\mathcal{X}_m \subset \mathcal{D}$
$\mathcal{P}$	partition of $\mathcal{D}$, $\mathcal{P} = \{\mathcal{X}_m\}_{m=1}^M$, with $\mathcal{X}_1 \cup \mathcal{X}_2 \cup ... \cup \mathcal{X}_M = \mathcal{D}$.
$\bar{\pi}(\mathbf{x})$	Normalized posterior pdf, $\bar{\pi}(\mathbf{x}) = p(\mathbf{x}\|\mathbf{y})$
$\pi(\mathbf{x})$	Unnormalized posterior function, $\pi(\mathbf{x}) = Z\bar{\pi}(\mathbf{x}) \propto \bar{\pi}(\mathbf{x})$
$\widehat{\pi}^{(N)}(\mathbf{x})$, $\widetilde{\pi}^{(N)}(\mathbf{x})$	Particle approximation of the measure of $\bar{\pi}(\mathbf{x})$ with N samples
$\mathbf{x}_n$	n-th Monte Carlo sample according to the pdf $\bar{\pi}(\mathbf{x})$
$\widetilde{\mathbf{x}}_{m,j}$	j-th Monte Carlo sample within $\mathcal{X}_m$
N	Number of Monte Carlo samples
M	Number of sub-regions $\{\mathcal{X}_m\}_{m=1}^M$
J_m	Number of Monte Carlo samples within $\mathcal{X}_m$
$\widehat{I}_L$, $\widetilde{I}_L$	Estimators of the integral $I = E_\pi[h(\mathbf{X})]$, using L samples
Z	Normalizing constant of $\pi(\mathbf{x})$ (marginal likelihood)
$\widehat{Z}$	Estimator of the normalizing constant Z
w_n, $\bar{w}_n$	Unnormalized and normalized IS weight of n-th sample
$\bar{\beta}_n$	$\bar{\beta}_n = \frac{1}{N}$ for equally-weighted samples, and $\bar{\beta}_n = \bar{w}_n$ for the IS scenario
$\bar{a}_m$	$\mathbb{P}(\mathbf{X} \in \mathcal{X}_m) = \int_{\mathcal{X}_m} \bar{\pi}(\mathbf{x}) d\mathbf{x}$
$\mathbf{s}_m$	Summary particle, $\mathbf{s}_m \in \mathcal{X}_m$, of the m-th sub-region $\mathcal{X}_m$
$\widehat{a}_m$	Summary weight of m-th summary particle, estimator of $\bar{a}_m$
W	Aggregate weight, associated to the discrete measure $\{\mathbf{s}_m, \widehat{a}_m\}_{m=1}^M$

Table II
SUMMARY OF DIFFERENT WEIGHTS AND THEIR RELATIONSHIPS.

Weights	**CUS**	**CIS**	**Property**	**Description**
$\bar{\beta}_n$	$\frac{1}{N}$	$\bar{w}_n = \frac{1}{N\widehat{Z}} \frac{\pi(\mathbf{x}_n)}{q(\mathbf{x}_n)}$	$\sum_{n=1}^N \bar{\beta}_n = 1$	MC weights
$\bar{\beta}_{m,j}$	$\frac{1}{N}$	$\bar{w}_{m,j} = \frac{1}{N\widehat{Z}} \frac{\pi(\widetilde{\mathbf{x}}_{m,j})}{q(\widetilde{\mathbf{x}}_{m,j})}$	$\sum_{m=1}^M \sum_{j=1}^{J_m} \bar{\beta}_{m,j} = 1$	MC weights
$\widehat{a}_m$	$\frac{J_m}{N}$	$\frac{J_m \widehat{Z}_m}{N\widehat{Z}}$	$\sum_{m=1}^M \widehat{a}_m = 1$	C-MC weights
a_m	J_m	$J_m \widehat{Z}_m$	$\sum_{m=1}^M a_m = W$	unnormalized C-MC weights
$\bar{\gamma}_{m,j}$	$\frac{1}{J_m}$	$\frac{1}{J_m \widehat{Z}_m} \frac{\pi(\widetilde{\mathbf{x}}_{m,j})}{q(\widetilde{\mathbf{x}}_{m,j})}$	$\sum_{j=1}^{J_m} \bar{\gamma}_{m,j} = 1$	MC weights in $\mathcal{X}_m$
W	N	$N\widehat{Z}$	—	C-MC aggregated weight
β_n equiv. to $\beta_{m,j}$, and $\beta_{m,j} = \widehat{a}_m \bar{\gamma}_{m,j}$. $\widehat{a}_m$ is an estimator of $\bar{a}_m = \mathbb{P}(\mathbf{X} \in \mathcal{X}_m)$.				

Supplementary Material Compressed Monte Carlo for Distributed Bayesian Inference

Luca Martino, Victor Elvira

I. Variance, bias and bound of the stratified estimators

Let us consider that K_m samples are drawn from each subregion, i.e., $\{\mathbf{s}_{m,k}\}_{k=1}^{K_m} \sim \bar{\pi}_m(\mathbf{x})$, for $m = 1, ..., M$. Then, the stratification estimator and approximation are, respectively,

$$\widetilde{I}_V = \sum_{m=1}^{M} \bar{a}_m \left[\frac{1}{K_m} \sum_{i=1}^{K_m} h(\mathbf{s}_{m,i}) \right],$$
$$\widetilde{\pi}^{(V)}(\mathbf{x}) = \sum_{m=1}^{M} \sum_{i=1}^{K_m} \frac{\bar{a}_m}{K_m} \delta(\mathbf{x} - \mathbf{s}_{m,i}),$$

where $\bar{a}_m = \int_{\mathcal{X}_m} \bar{\pi}(x)dx = \frac{Z_m}{Z}$, $V = \sum_{m=1}^{M} K_m$ is the total number of generated samples. The stratified estimator is unbiased,

$$E_{\bar{\pi}}[\widetilde{I}_V] = \sum_{m=1}^{M} \bar{a}_m \left[\int_{\mathcal{X}_m} h(\mathbf{x}) \bar{\pi}_m(\mathbf{x}) d\mathbf{x} \right], \quad (1)$$
$$= \sum_{m=1}^{M} \bar{a}_m I_m = I, \quad (2)$$

with variance

$$\text{Var}_{\bar{\pi}} \left[\widetilde{I}_V \right] = \sum_{m=1}^{M} \frac{1}{K_m} \bar{a}_m^2 \sigma_m^2, \quad (3)$$

where $\sigma_m^2 = \text{var}_{\bar{\pi}_m}[h(\mathbf{X})] = \int_{\mathcal{X}_m} (h(\mathbf{x}) - I_m)^2 \bar{\pi}_m(\mathbf{x}) d\mathbf{x}$, and $I_m = E_{\bar{\pi}_m}[h(\mathbf{X})] = \int_{\mathcal{X}_m} h(\mathbf{x}) \bar{\pi}_m(\mathbf{x}) d\mathbf{x}$, i.e., the variance and mean of the random variable $h(\mathbf{X})$ when $\mathbf{X}$ is restricted within $\mathcal{X}_m$ [1]. Note that if $K_m = K$ for all m, hence $V = MK$, then

$$\text{Var}_{\bar{\pi}} \left[\widetilde{I}_V \right] = \frac{1}{K} \sum_{m=1}^{M} \bar{a}_m^2 \sigma_m^2. \quad (4)$$

A. Variance of $h(\mathbf{X})$

Let us recall the definition of the restricted target pdf, $\bar{\pi}_m(\mathbf{x}) = \frac{1}{\bar{a}_m} \bar{\pi}(\mathbf{x}) \mathbb{I}(\mathcal{X}_m) = \frac{1}{Z_m} \pi(\mathbf{x}) \mathbb{I}(\mathcal{X}_m)$. An interesting expression of the variance of the random variable $h(\mathbf{X})$ can be found below. Indeed, it can expressed as sum of two terms: the first one considering of the variances within each the subregions,

$$\sigma_m^2 = \text{var}_{\bar{\pi}_m}[h(\mathbf{X})] = \int_{\mathcal{X}_m} (h(\mathbf{x}) - I_m)^2 \bar{\pi}_m(\mathbf{x}) d\mathbf{x},$$
$$I_m = E_{\bar{\pi}_m}[h(\mathbf{X})] = \int_{\mathcal{X}_m} h(\mathbf{x}) \bar{\pi}_m(\mathbf{x}) d\mathbf{x}, \quad (5)$$

and the second one considering the variance between the subregions, i.e., $\sum_{m=1}^{M} \bar{a}_m (I_m - I)^2$. Namely, we have

$$\sigma^2 = \text{var}_{\bar{\pi}}[h(\mathbf{X})] = \int_{\mathcal{D}} (h(\mathbf{x}) - I)^2 \bar{\pi}(\mathbf{x}) d\mathbf{x}, \quad (6)$$
$$= \sum_{m=1}^{M} \bar{a}_m \sigma_m^2 + \sum_{m=1}^{M} \bar{a}_m (I_m - I)^2, \quad (7)$$

where we have used the equality $\text{var}_{\bar{\pi}}[h(\mathbf{X})] = E_{\bar{\pi}}[\text{var}_{\bar{\pi}_m}[h(\mathbf{X})]] + \text{var}_{\bar{\pi}}[E_{\bar{\pi}_m}[h(\mathbf{X})]]$ [1]. As a consequence, we can also write

$$\sigma^2 \geq \sum_{m=1}^{M} \bar{a}_m \sigma_m^2. \quad (8)$$

The result above is valid for any kind of partition.

B. Bound for the approximation

For the sake of simplicity, let assume $d_X = 1$, $\mathcal{D} = [e_1, e_2]$ and a bounded target $\bar{\pi}$. Let us consider a uniform grid partition with step κ forming M intervals, so that $\kappa = \frac{e2-e1}{M} = \frac{\Delta}{M}$ where $\Delta = e_2 - e_1$. Then, consider the following particle approximation

$$\widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^{M} \bar{a}_m \delta(x - s_m), \quad (9)$$

obtained by stratification. Considering one fixed realization and a sorted sequence $s_1 \leq s_2 \leq ... \leq s_m$. Then, define $\nu = \max |s_i - s_{i+1}|$ with $i = 1, ..., M - 1$. The particle approximation (9) corresponds to a piecewise constant approximation of the cumulative function $F_X(x) = \int_{e_1}^{x} \bar{\pi}(z)dz$ of X. Therefore, considering the bound for a piecewise constant approximation, we can write

$$\max |F_X(x) - \widehat{F}_X(x)| \leq \frac{\nu \Delta ||\bar{\pi}||_\infty}{2M} \leq \frac{\Delta^2 ||\bar{\pi}||_\infty}{2M} = \frac{c}{M}, \quad (10)$$

where $||\bar{\pi}||_\infty = \max \bar{\pi}(x)$ and $c = \frac{1}{2}\Delta^2 ||\bar{\pi}||_\infty$ [**?**]Recall that, with a standard MC approximation, the bound is of order $\frac{1}{\sqrt{M}}$.

REFERENCES

[1] A. Owen. *Monte Carlo theory, methods and examples*. http://statweb.stanford.edu/∼owen/mc/, 2013.