AN ADAPTIVE POPULATION IMPORTANCE SAMPLER: LEARNING FROM UNCERTAINTY

Luca Martino[⋆]*, Víctor Elvira*[†]*, David Luengo*[‡]*, Jukka Corander*[⋆]

[⋆] Dep. of Mathematics and Statistics, University of Helsinki, 00014 Helsinki (Finland).
[†] Dep. of Signal Theory and Communic., Universidad Carlos III de Madrid, 28911 Leganés (Spain).
[‡] Dep. of Circuits and Systems Engineering, Universidad Politécnica de Madrid, 28031 Madrid (Spain).

ABSTRACT

Monte Carlo (MC) methods are well-known computational techniques widely used in different fields such as signal processing, communications and machine learning. An important class of MC methods is composed of importance sampling (IS) and its adaptive extensions, e.g., Adaptive Multiple IS (AMIS) and Population Monte Carlo (PMC). In this work, we introduce a novel adaptive and iterated importance sampler using a population of proposal densities. The proposed algorithm, named *Adaptive Population Importance Sampling* (APIS), provides a global estimation of the variables of interest iteratively, making use of all the samples previously generated. APIS combines a sophisticated scheme to build the IS estimators (based on the deterministic mixture approach) with a simple temporal adaptation (based on epochs). In this way, APIS is able to keep all the advantages of both AMIS and PMC while minimizing their drawbacks. Futhermore, the cloud of proposals is adapted in such a way that local features of the target density can be better taken into account compared to single global adaptation procedures. The result is a fast, simple, robust and high-performance algorithm applicable to a wide range of problems. Numerical results show the advantages of the proposed sampling scheme for a toy example and a localization problem in a wireless sensor network.

***Index Terms*—** Monte Carlo methods, adaptive importance sampling, population Monte Carlo, iterative estimation.

1. INTRODUCTION

Monte Carlo (MC) methods are widely used in signal processing and communications for statistical inference and stochastic optimization [1, 2, 3, 4, 5, 6, 7]. Importance sampling (IS) [8, 9] is a well-known MC methodology to compute efficiently integrals involving a complicated multidimensional target probability density function (pdf), $\pi(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^n$. Moreover, it is often used in order to calculate the normalizing constant of $\pi(\mathbf{x})$ (also called *partition function*) [9], which is required in several applications, like model selection [10, 11, 12].

The IS technique draws samples from a simple proposal pdf, $q(\mathbf{x})$, assigning weights to them according to the ratio between the target and the proposal, i.e., $w(\mathbf{x}) = \frac{\pi(\mathbf{x})}{q(\mathbf{x})}$. However, although the validity of this approach is guaranteed under mild assumptions, the variance of the estimator depends notably on the discrepancy between the shape of the proposal and the target. Markov Chain Monte Carlo (MCMC) algorithms [8, 13] are another well-known class of MC techniques which generate a Markov chain converging to the target distribution. MCMC techniques often lead to random walks of the samples generated around the regions of high probability. This explorative behaviour is responsible for MCMC methods being usually preferred in high-dimensional applications [6, 14, 15, 16]. Nevertheless, MCMC algorithms also suffer from several important shortcomings [8, 9]: the diagnostic of the convergence is often difficult and it is not straightforward to estimate the partition function (i.e., the normalizing constant of the target) given the generated samples.

In order to overcome these issues, substantial effort has been devoted to the design of adaptive IS schemes [9], where the proposal density is updated by learning from all the previously generated samples. The Population Monte Carlo (PMC) [17] and the Adaptive Multiple Importance Sampling (AMIS) [18] methods are two general schemes that combine the proposal adaptation idea with the cooperative use of a cloud of proposal pdfs. On the one hand, in PMC a population of proposals is updated using propagation and resampling steps [9, Chapter 14]. The IS estimator is built as in the standard IS approach, but using a mixture of different proposals [19, 20, 21, 22]. PMC schemes have been widely used in signal processing applications due to their simplicity and flexibility [22, 23, 24].

This work has been supported by the Spanish government's projects COMONSENS (CSD2008-00010), ALCIT (TEC2012-38800-C03-01), DISSECT (TEC2012-38058-C03-01) and COMPREHENSION (TEC2012-38883-C02-01), and by the ERC grant 239784 and AoF grant 251170.

On the other hand, in AMIS a single proposal is adapted as in a standard adaptive IS fashion, but the sequence of all the previous proposals is used to build the importance weights and global estimator according to the so-called *deterministic mixture* approach [25, 26]. This implies that all the previous proposals must be evaluated at the new samples, and also the new proposal pdf must be evaluated at all the previous samples, thus yielding an increase in computational cost as the algorithm evolves in the time. This single proposal could also be a mixture of pdfs, but the adaptation in this case involves more complicated methodologies (such as clustering), increasing the computational cost even more [27]. AMIS is not as well-known as PMC, but has been successfully applied to genetic inference problems recently [28].

Finally, let us remark that the update of the proposals in both methodologies (AMIS and PMC) can also be carried out according to some optimality criterion, such as the minimization of the Kullback-Leibler divergence [19, 20, 21], although at the expense of an increased complexity. Thus, they seems to present certain instability in the adaptation, in general.

In this work, we introduce a novel population scheme, *adaptive population importance sampling* (APIS).[1] APIS draws samples from different proposal densities at each iteration, weighting these samples according to the deterministic mixture approach [25, 26] for a fixed (i.e., non-adaptive) setting. At each iteration, the APIS algorithm computes iteratively a *global* IS estimator, taking into account all the generated samples up to that point.

The main difference w.r.t. the existing AMIS and PMC schemes lies in the more streamlined adaptation procedure of APIS, as well as in the approach followed to build the estimators. APIS starts with a cloud of N proposals initialized randomly or according to the prior information available, with the location parameters spread in the state space. Each proposal can use a different scale parameter. Since the adaptation of the scale parameters can be an issue for the performance of the sampler, we focus on the update of the location parameters in order to ensure the robustness of the algorithm. The algorithm is then divided into groups (*epochs*) of T_a iterations, where the proposals are kept fixed and T_a samples are drawn from each one. At the end of every epoch, the T_a samples drawn from each proposal are used to update its parameters (using *partial IS estimators*). At each transition between epochs, the location parameters are updated and the adaptation memory "refreshed". This approach enables APIS to learn from the uncertainty associated to each proposal, as evidenced by the variance of the different partial IS estimators. Furthermore, it allows each proposal to concentrate on some particular region of the state space, thus modelling specific and localized features of the target. In this way, APIS can obtain a very good global approximation of the target by combining all the local approximations, as long as different proposals explore different parts of the state space. Finally, note that this is achieved without any additional computation in terms of evaluation of the target and proposal pdfs.

Unlike PMC, the novel technique does not require resampling steps to prevent the degeneracy of the mixture, thus avoiding the loss of diversity in the population. This is a common problem for sampling-importance resampling type algorithms, where additional MCMC moves are occasionally applied [30]. Indeed, in [31] the authors attempt to diminish this negative effect by forcing "artificially" a pre-defined amount of the highest importance weights to be equal to control the loss of diversity caused by resampling. Following the previous observations, we also propose a possible interaction among the proposal locations applying MCMC moves, preserving more the diversity in the population than the use of resampling steps. We call the resulting technique *Markov APIS (MAPIS)*. MAPIS contains two sources of movement: the APIS movements with the addition of MCMC iterations.

In APIS, at each iteration, the cloud of proposals partake jointly in the construction of an IS estimator using the deterministic mixture approach [25, 26], that introduces more stability in the estimation. This estimator is combined with the past estimators using a simpler strategy than in AMIS: a standard (simpler than the determinist mixture) IS estimator using multiple proposals is built. Therefore, in this sense, APIS follows an approach "in-between" PMC and AMIS (for further clarifications see Appendix 9 in this work). Numerical results show that APIS improves the performance of a standard non-adaptive multiple importance sampler regardless of the initial conditions and parameters. We have also compared the performance of APIS to that of several AMIS and PMC schemes, showing that APIS outperforms both approaches in terms of robustness to the choice of the initial parameters.

The paper is organized as follows. The general problem statement is described in Section 2, and the novel APIS algorithm is introduced in Section 3. In Section 4, we discuss how to tune efficiently the parameters of the novel scheme and suggest a more robust implementation. In Section 5, we introduce the additional MCMC adaptation which leads to the MAPIS algorithm. Section 6 is devoted to analyzing differences and similarities among APIS, AMIS and PMC methods. An exhaustive numerical comparison among these three methods is then provided in Section 7, whereas the application of APIS to a localization problem in wireless sensor networks [32, 33, 34] is considered in Section 8. Finally, we conclude with a brief summary in Section 9.

[1] A preliminary version of this work has been published in [29]. With respect to that paper, here we propose an interacting adaptation using an MCMC technique, discuss the construction and the consistency of the estimators, and provide more exhaustive numerical simulations for the toy example, as well as a localization example in wireless sensor networks. Comparisons with other sampling algorithms are also included.

2. PROBLEM STATEMENT AND AIM OF THE WORK

In many applications, we are interested in inferring a variable given a set of observations or measurements. Let us consider the variable of interest, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$, and let $\mathbf{y} \in \mathbb{R}^d$ be the observed data. The posterior pdf is then

$$p(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{Z(\mathbf{y})} \propto \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}), \tag{1}$$

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 or partition function (useful in model selection). In general, $Z(\mathbf{y})$ is unknown, so we consider the corresponding (usually unnormalized) target pdf,

$$\pi(\mathbf{x}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}). \tag{2}$$

Our goal is computing efficiently the expected value of $f(\mathbf{X})$, where $\mathbf{X} \sim \frac{1}{Z}\pi(\mathbf{x})$, i.e., an integral measure w.r.t. the target pdf,

$$I = E[f(\mathbf{X})] = \frac{1}{Z}\int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \tag{3}$$

where $Z = \int_{\mathcal{X}} \pi(\mathbf{x})d\mathbf{x}$. Our goal is to design a sampling algorithm able to estimate jointly I and Z. Furthermore, we would like to obtain a sampler as efficient and robust as possible, so that the interested user can apply it easily to different problems without having to perform an exhaustive fine tuning of the proposed approach.

3. THE APIS ALGORITHM

The adaptive population importance sampling (APIS) algorithm estimates jointly Z and I by drawing samples from a population of adaptive proposals. For the sake of simplicity, here we only consider a population of *Gaussian* proposal pdfs. However, the underlying idea is more general: many kinds of proposals could be used, including mixtures of different types of proposals.

The covariance matrices of the proposals are kept fixed. Hence, the adaptation of APIS focuses on the location parameters (the mean of the Gaussians in this case). We have decided not to adapt higher order parameters (unlike [16, 35]), in order to reinforce the robustness of the sampler: the adaptation of higher order parameters is very delicate and can compromise the performance of the algorithm. However, in contrast to AMIS, in APIS we can use different scale parameters for the cloud of proposals. This allows us to strongly reduce the dependence on the initial conditions without affecting the perfomance or increasing the computational cost. Note also that resampling steps are not used in order to avoid the loss of diversity in the population. The APIS algorithm for Gaussian proposal pdfs is summarized in Table 1.

3.1. Remarks and observations

In this section, we provide some important remarks on several aspects of the APIS algorithm. First of all, let us observe that the algorithm works on two different time scales:

• At each iteration ($t = 1, \ldots, T = MT_a$), APIS computes the "current" estimate of the desired integral, $\hat{J}_t$, and updates recursively the global estimates of the desired integral and the normalizing constant, $\hat{I}_t$ and $\hat{Z}_t$ respectively.

• At the end of each epoch, the parameters of the N proposals, $\boldsymbol{\mu}_i^{(m)}$ for $1 \leq i \leq N$, are updated.

Moreover, note that:

1) All the different proposal pdfs must be normalized.

2) We can equivalently state that, in APIS, we draw from a mixture of N pdfs, with equal weights

$$q^*(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N} q_j^{(m)}(\mathbf{z}_i), \tag{12}$$

as shown in the denominator in Eq. (4). The way to draw from this mixture is called "deterministic" [25, 26] since we draw always one sample from each pdf in the mixture. For further details, see Appendix .2. APIS provides a procedure to update the location parameters μ_i, $i = 1, \ldots, N$, in the mixture.

3) The update of the location parameters $\mu_i^{(m)}$ is done only using the last T_a samples drawn from the i-th proposal, and building partial IS estimators of the expected value of the target. APIS is driven by the "IS estimation errors" since it takes advantage of the variances of each partial IS estimators. The memoryless feature of APIS facilitates that each proposal pdf could describe

Table 1. **APIS algorithm.**

1. **Initialization:** Set $t = 1$, $m = 0$, $\hat{I}_0 = 0$ and $L_0 = 0$. Choose N *normalized* Gaussian proposal pdfs,

$$q_i^{(0)}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^{(0)}, \mathbf{C}_i), \qquad i = 1, \ldots, N,$$

with mean vectors $\boldsymbol{\mu}_i^{(0)}$ and covariance matrices $\mathbf{C}_i$ ($i = 1, \ldots, N$). Select the number of iterations per epoch, $T_a \geq 2$, and the total number of iterations, $T = MT_a$, with $M \leq \frac{T}{2} \in \mathbb{Z}^+$ denoting the number of epochs. Set also $\boldsymbol{\eta}_i = \mathbf{0}$ and $W_i = 0$ for $i = 1, \ldots, N$.

2. **IS steps:**

   (a) Draw $\mathbf{z}_i \sim q_i^{(m)}(\mathbf{x})$ for $i = 1, \ldots, N$.

   (b) Compute the importance weights,

$$w_i = \frac{\pi(\mathbf{z}_i)}{\frac{1}{N}\sum_{j=1}^{N} q_j^{(m)}(\mathbf{z}_i)}, \quad i = 1, \ldots, N, \tag{4}$$

   and normalize them, $\bar{w}_i = \frac{w_i}{S}$, where $S = \sum_{j=1}^{N} w_j$.

3. **Iterative IS estimation:** Calculate the "current" estimate of $I = E[f(\mathbf{X})]$,

$$\hat{J}_t = \sum_{i=1}^{N} \bar{w}_i f(\mathbf{z}_i) \approx I, \tag{5}$$

   and the *global estimate*, using the recursive formula

$$\hat{I}_t = \frac{1}{L_{t-1} + S}\left(L_{t-1}\hat{I}_{t-1} + S\hat{J}_t\right) \approx I, \tag{6}$$

   where $L_t = L_{t-1} + S$. Note that $\hat{Z}_t = \frac{1}{Nt} L_t$.

4. **Learning:**

   (a) Compute

$$\rho_i = \frac{\pi(\mathbf{z}_i)}{q_i^{(m)}(\mathbf{z}_i)}, \qquad i = 1, \ldots, N. \tag{7}$$

   (b) Calculate the *partial* estimations of the mean of the target,

$$\boldsymbol{\eta}_i = \frac{1}{W_i + \rho_i}\left(W_i\boldsymbol{\eta}_i + \rho_i\mathbf{z}_i\right) \approx E[\mathbf{X}], \tag{8}$$

   and set $W_i = W_i + \rho_i$ for $i = 1, \ldots, N$.

5. **Proposal adaptation:** If $t = kT_a$ ($k = 1, 2, \ldots, M$):

   (a) Change the location parameters $\boldsymbol{\mu}_i^{(m)}$ according to their partial estimations of the mean of the target, i.e., set

$$\boldsymbol{\mu}_i^{(m+1)} = \boldsymbol{\eta}_i, \qquad i = 1, \ldots, N, \tag{9}$$

   and $q_i^{(m+1)} = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^{(m+1)}, \mathbf{C}_i)$.

   (b) "Refresh memory" by setting $\boldsymbol{\eta}_i = \mathbf{0}$ and $W_i = 0$ for $i = 1, \ldots, N$. Set also $m = m + 1$.

6. **Stopping rule:** The simplest possibility is: If $t < T$, set $t = t + 1$ and repeat from step 2. Otherwise, end.

7. **Outputs:** Return the estimate of the desired integral,

$$\hat{I}_T \approx I = \frac{1}{Z}\int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \tag{10}$$

   as well as the normalizing constant of the target pdf,

$$\hat{Z}_T \approx Z = \int_{\mathcal{X}} \pi(\mathbf{x})d\mathbf{x}. \tag{11}$$

local features of the target. Typically, the proposals remain invariably in some regions or a random walk is generated around areas of high probabilities, in the state space $\mathcal{X} \subset \mathbb{R}^n$.

4) Steps 4 and 5 of APIS do not require additional evaluations of the target and the proposal pdfs since they are already evaluated at $\mathbf{z}_i$, $i = 1, \ldots, N$, in step 2.

5) The global estimators, $\hat{I}_T$ and $\hat{Z}_T$, are iteratively obtained by an importance sampling approach using NT samples drawn from NM different proposals: N initial proposals chosen by the user, and $N(M-1)$ proposals adapted by the algorithm.

6) Different stopping rules can be applied to ensure that the global estimators produce the desired degree of accuracy, in terms of Monte Carlo variability. For instance, one possibility is taking into account the variation of the estimate over time. In this case, the algorithm could be stopped at any iteration $t^* < T$, since an IS approach does not have the convergence issues ("burn-in" period) appearing in MCMC methods.
7) *PIS scheme (static APIS)*: A degenerate static algorithm appears if $T_a = T$ (i.e., $M = 1$): indeed, in this case, the adaptation is not applied. In this static scenario an iterated multiple IS algorithm is performed, denoted as PIS or static APIS. PIS combines the deterministic mixture idea and the standard IS approach to build the global estimators and, for this reason, is different from a standard multiple IS scheme. For more details, see Appendix 9.

Finally, consider only the transitions (i.e., $t = mT_a$), In APIS, we draw $T_a \geq 2$ samples from each proposal, providing a single global estimation. Thus, in the previous description the index t could be removed. Indeed, *within an epoch* the proposals do not change, so we could draw T_a i.i.d. samples directly from each proposal and then adapt the proposals using these samples. However, we prefer to maintain the previous description to emphasize the fact that the accuracy of the estimator can be tested at each iteration t, and that the algorithm could be stopped at any time.

3.2. Blind adaptation

APIS considers a completely blind strategy to change the values of $\boldsymbol{\mu}_i$. This is an important feature in order to design a robust and light algorithm, i.e., (a) less dependent on the choice of the parameters, (b) less costly, and easier to be implemented. In this approach, the algorithm attempts to improve the positions $\boldsymbol{\mu}_i$ only using the evaluations of the target achieved online. Indeed, we are not considering any discrepancy function between the target and the proposals or any other optimal criteria for decreasing the variance of the IS estimator. Using this blind approach, a good final configuration of the location parameters is to be around all the modes of the target. Thus, in this case, an ideal configuration is that $\boldsymbol{\mu}_i$ be exactly distributed as the target π. This behaviour can be also found in the standard PMC schemes where the location parameters are adapted using resampling steps unlike in APIS, e.g., discarding means $\boldsymbol{\mu}_i$ far away from the modes of π and replacing the well-located means $\boldsymbol{\mu}_i$ (according to the importance weights). The APIS adaptation yields similar effects to the use of resampling but reducing the loss of diversity in the population.

Fig. 1 shows a contour plot of a multimodal target $\pi(\mathbf{x})$ and the evolution of the location parameters $\boldsymbol{\mu}_i$, after a run of $T = 2000$ iterations of APIS with $M = 40$, $T_a = \frac{T}{M} = 50$ (using Gaussian proposals with $\mathbf{C}_i = \sigma^2\mathbf{I}_2$, $\sigma = 5$). In Fig. 1(b), the initialization ($\boldsymbol{\mu}_i^{(0)}$, shown with squares) is obviously better then in Fig. 1(a), covering the areas of high probability. Fig. 1 also depicts the final locations of the means, $\boldsymbol{\mu}_i^{(T)}$ (circles), after T iterations of APIS. Furthemore, the trajectories of two means in the population are depicted by a dashed line. Note that, in some cases, a random walk among different modes is induced whereas, in other cases, the corresponding mean remains trapped in a local mode, after some iterations. For further considerations, analysis and details, see Section 5.

4. CHOICE OF THE PARAMETERS AND ROBUST IMPLEMENTATION

As in any other Monte Carlo technique, the performance of APIS depends on the initialization and choice of the parameters, although this sensitivity is reduced w.r.t. a standard IS approach, as illustrated in the simulations. Hence, if some prior information about the target is available, it should be used to choose the initial parameters. However, if no prior information is available, a *black-box* and *robust* implementation is desired by the user. In the following, we discuss how changes in the parameters affect the performance and, at the same time, provide some suggestions to design an APIS algorithm as robust as possible.
• *Class of proposal pdfs:* The class of the proposal densities, i.e., their parametric form, can be different for each proposal. In this work, for the sake of simplicity, we focus our attention on the use of Gaussian proposal pdfs. Obviously, we can use different type of proposal densities, or also a mixture of different kinds of pdfs. For the black-box and robust implementation point of view, a better choice is the use of heavy-tailed distributions as t-Students pdfs, for instance. This is also the idea in the so-called *defensive sampling* strategy in [36], [9, Chapter 3].
• *Initial location parameters $\boldsymbol{\mu}_i^{(0)}$:* If no information of the target is provided, then the values of $\mu_i^{(0)}$, $i = 1, \ldots, N$, should be chosen in order to cover as much as possible of the target domain, $\mathcal{X} \subseteq \mathbb{R}^n$ (in analogy to using an improper uniform prior).
• *Scale parameters $\mathbf{C}_i$:* The adaptation of moments of higher order is, in general, more challenging and crucial for performance. For instance, an inadequate adjustment of the scale parameters, at some iteration t, can easily jeopardize the capability of the sampler for the next iterations. As a consequence, in general, the adaptation of the scale parameters needs an accurate theoretical study and a thorough tuning of the initial value of parameters by the user, before applying the sampler. For this reason, in this work, we prefer to adapt only the location parameters of proposals since APIS allows the joint use of several variances. The simplest possibility is to choose randomly within a range the scale parameters of each proposal. Another possibility is to

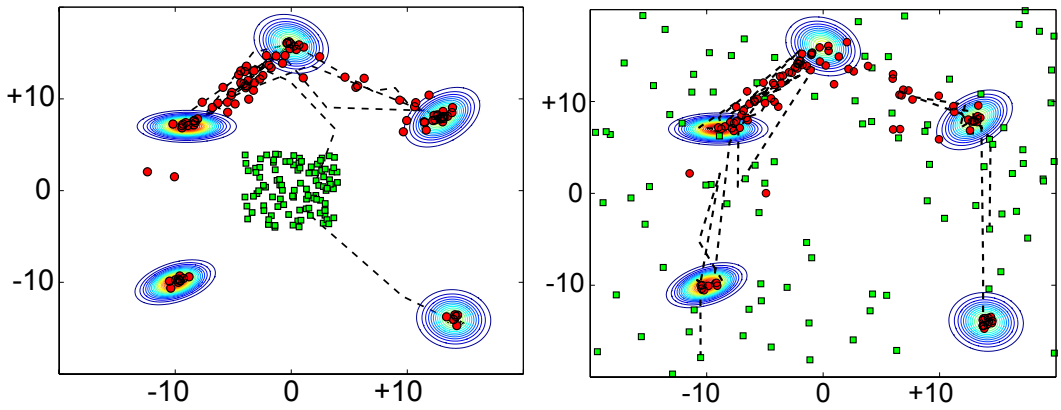

Fig. 1. Contour plot of the target $\pi(\mathbf{x})$ in Section 7.1, the initial means $\boldsymbol{\mu}_i^{(0)}$ (squares) and the final means $\boldsymbol{\mu}_i^{(T)}$ (circles) locations of the means of the proposals for a single run of APIS ($\sigma = 5$, $N = 100$, $M = 40$, $T = 2000$, $T_a = \frac{T}{M} = 50$). The trajectories of two means in the sample population are depicted in dashed line. **(a)** "Bad" initialization denoted as **In1**, $\boldsymbol{\mu}_i^{(0)} \sim \mathcal{U}([-4, 4] \times [-4, 4])$. **(b)** A better initialization denoted as **In2**, $\boldsymbol{\mu}_i^{(0)} \sim \mathcal{U}([-20, 20] \times [-20, 20])$.

associate more than one variance to each proposal pdf. Indeed, consider N_μ initial location parameters $\boldsymbol{\mu}_i^{(0)} \in \mathcal{X} \subseteq \mathbb{R}^n$. For each $\boldsymbol{\mu}_i^{(0)}$, choose N_σ different scale parameters, implying that the total number of different proposals is $N = N_\mu N_\sigma$. A further possible idea is to divide the proposal pdfs in two groups: "sedentary" proposals for local search (small variances) and "nomad" proposals with explorative behaviour (bigger variances).

• *Choice of* $T_a = \frac{T}{M}$*:* The parameter T_a is the number of iterations in each epoch, i.e., the number of iterations with the same location parameters $\boldsymbol{\mu}_i$. As a consequence, T_a is also the number of samples used to choose the new location parameters, at the end of the epoch. As T_a grows, each partial IS estimator $\boldsymbol{\eta}_i$ (used to adapt the the proposals) would provide a better estimation, closer to the expect value of the target $\pi(\mathbf{x})$, and also closer to each others, $i = 1, \dots, N$. Although it is clearly a good scenario, paradoxically in APIS, it is not the best situation. Indeed, the proposals tend to cover the same region of the target domain, losing diversity in the population. When T_a becomes smaller, the proposal pdfs tend to be spread out around the regions of high probability, that could be better configuration for the performance of the global estimator. If T_a is too small, large and almost random movements of the proposals are facilitated so that some proposal could yield an explorative random walk in the state space[2]. However, even with a bad choice of the parameters, APIS remains in any case a valid technique: even in the worst cases, APIS provides similar behaviours and better performance than an adaptive IS scheme (using a single proposal) and a static multiple IS scheme with a random choice of the parameters.

The numerical results in Section 7 also suggest the existence of an optimal value T_a^* which depends of the scale parameters of the proposals. In general, the use of small variances prefers a use of smaller valuse of T_a, whereas big variances prefer greater values of T_a. Different values of $T_a^{(i)}$, one for each proposal, could be applied according to the scale parameter of the i-th proposal. Furthermore, the values $T_a^{(i)}$ can be also changed with the time step: smaller values for a more explorative behaviour at the beginning, and greater values to reduce the "entropy" of the clouds.

4.1. Sequential approach

In order to increase the robustness of the performance of APIS, a sequential approach can be used. Indeed, as the dimension n of the space $\mathcal{X} \in \mathbb{R}^n$ increases, the inference problems are intensified, since the relative volume of $\mathbf{x}$ where $\pi(\mathbf{x})$ is high becomes extremely small. Consider to split the data $\mathbf{y}$ into R different disjoint sets, i.e., $\mathbf{y} = \cup_{r=1}^R \mathbf{y}^{(r)}$ and $\mathbf{y}^{(j)} \cap \mathbf{y}^{(k)} = \emptyset$ for all $j, k \in \{1, \dots, R\}$. Then, we apply APIS starting with the initial cloud $\{\boldsymbol{\mu}_{i,1}^{(0)}\}_{i=1}^N$, and considering the partial (unnormalized) target pdf

$$\pi_1(\mathbf{x}) = \ell(\mathbf{y}^{(1)}|\mathbf{x})g(\mathbf{x}),$$

[2]A physical analogy could help the reader in understanding the behavior: defining an *energy* variable $E = \frac{1}{T_a}$, increasing T_a means to cool down the cloud (less energy E) whereas decreasing T_a means to heat up the system, i.e., rise the energy E in the cloud of particles, increasing the total entropy.

for T iterations and $M = \frac{T}{T_a}$ epochs (T can be relatively small). Thus we obtain a final configuration of the cloud $\{\boldsymbol{\mu}_{i,1}^{(M)}\}_{i=1}^N$. Then, we apply again APIS starting with $\{\boldsymbol{\mu}_{i,2}^{(0)}\}_{i=1}^N = \{\boldsymbol{\mu}_{i,1}^{(M)}\}_{i=1}^N$ and considering the partial (unnormalized) target pdf

$$\pi_2(\mathbf{x}) = \ell(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}|\mathbf{x})g(\mathbf{x}) = \ell(\mathbf{y}^{(1:2)}|\mathbf{x})g(\mathbf{x}),$$

and so on. At the r-th step, we start with $\{\boldsymbol{\mu}_{i,r}^{(0)}\}_{i=1}^N = \{\boldsymbol{\mu}_{i,r-1}^{(M)}\}_{i=1}^N$ and handle the target $\pi_r(\mathbf{x}) = \ell(\mathbf{y}^{(1:r)}|\mathbf{x})g(\mathbf{x})$. Note that $\pi_R(\mathbf{x}) = \pi(\mathbf{x})$. This strategy helps the convergence of the estimators, yielding more robust performance. There is a clear connection between this approach and the tempering of the target idea suggested in different works [37], where a sequence of different targets is used. However, we consider this strategy more natural since appears as an automatic "tempering", instead that changing the scale of target artificially. This sequential approach is also more informative since at each sub-step, APIS provides also an estimator $\hat{I}_T^{(r)}$ related to the quantity of interest.

5. MCMC INTERACTION: MARKOV APIS

In APIS the adaptation of the location parameter of a proposal is done independently from the rest of the population. Here, we propose a possible interaction procedure among the location parameters of the proposal pdfs (the means of the Gaussians). The easiest way could be to use a kind of resampling step: draw N samples from a multinomial probability mass function (pmf) using probabilities proportionally to $\pi(\boldsymbol{\mu}_i^{(m)})$. However, this procedure can yield a loss of diversity in the population. For this reason, we apply other approach described below.

We propose to share information applying an MCMC technique over the cloud of means $\boldsymbol{\mu}_i$, at each transition iterations between two epochs ($t = mT_a$ with $m = 1, \ldots, M$), or only in few of them. A possible adequate technique is the *Sample Metropolis-Hastings algorithm* [13, Chapter 5]: the underlying idea is to improve the positions of the means $\boldsymbol{\mu}_i$ and, as a consequence, to help the adaptation in the APIS algorithm. We will call the APIS algorithm using also SMH as *Markov APIS (MAPIS)*.

These MCMC iterations are applied after the step 5 of APIS. Thus, MAPIS contains both sources of movement: step 5 of APIS, with the additions of SMH iterations. Observe that, unlike the steps 4-5 of APIS, these SMH steps need also new evaluations of the target pdf. In the following, for the sake of simplicity, we remove the super-index in $\boldsymbol{\mu}_i$ denoting the current epoch.

5.1. Sample Metropolis-Hastings (SMH) algorithm

Consider the extended target pdf

$$\pi_g(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N) \propto \prod_{i=1}^N \pi(\boldsymbol{\mu}_i),$$

where each marginal $\pi(\boldsymbol{\mu}_i)$, $i = 1, ..., N$, coincides with the true target pdf ($\boldsymbol{\mu}_i \in \mathcal{X} \subseteq \mathbb{R}^n$). Let us denote $\tau = 1, \ldots, \Upsilon$ the SMH iteration index. At the τ-iteration, we consider *the population* of samples

$$\mathcal{P}_\tau = \{\boldsymbol{\mu}_{1,\tau}, ..., \boldsymbol{\mu}_{N,\tau}\}.$$

At each iteration, the underlying idea of SMH is to replace one "bad" sample in the population with a better one, according to certain suitable probabilities. The algorithm is designed so that, after a burn-in period τ_b, the elements in $\mathcal{P}_{\tau'}$ ($\tau' > \tau_b$) are distributed according to $\pi_g(\boldsymbol{\mu}_{1,\tau'}, \ldots, \boldsymbol{\mu}_{N,\tau'})$, i.e., $\boldsymbol{\mu}_{i,\tau'}$ are i.i.d. samples from $\pi(\mathbf{x})$. For $\tau = 1, ..., \Upsilon$, the SMH algorithm consists of the following steps:

1) Draw $\boldsymbol{\mu}_{0,\tau} \sim \varphi(\boldsymbol{\mu})$, where φ is another proposal density, chosen by the user, which could indeed be selected or adapted using the information obtained in the previous steps of APIS.

2) Choose a "bad" sample $\boldsymbol{\mu}_{k,\tau}$ from the population (i.e., $k \in \{1, ..., N\}$), according to a probability proportional to $\frac{\varphi(\boldsymbol{\mu}_{k,\tau})}{\pi(\boldsymbol{\mu}_{k,\tau})}$, which corresponds to the inverse of importance sampling weights.

3) Accept the new population

$$\mathcal{P}_{\tau+1} = \{\boldsymbol{\mu}_{1,\tau+1} = \boldsymbol{\mu}_{1,\tau}, ..., \boldsymbol{\mu}_{k,\tau+1} = \boldsymbol{\mu}_{0,\tau}, ..., \boldsymbol{\mu}_{N,\tau+1} = \boldsymbol{\mu}_{N,\tau}\},$$

with probability

$$\alpha(\boldsymbol{\mu}_{1,\tau}, ..., \boldsymbol{\mu}_{N,\tau}, \boldsymbol{\mu}_{0,\tau}) = \frac{\sum_{i=1}^N \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})}}{\sum_{i=0}^N \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})} - \min\limits_{0 \le i \le N} \frac{\varphi(\boldsymbol{\mu}_{i,\tau})}{\pi(\boldsymbol{\mu}_{i,\tau})}}.$$

Otherwise, set $\mathcal{P}_{\tau+1} = \mathcal{P}_\tau$. **4)** If $\tau < \Upsilon$, set $\tau = \tau + 1$ and repeat from step 1.
Observe that the difference between $\mathcal{P}_\tau$ and $\mathcal{P}_{\tau+1}$ is at most one sample. The ergodicity can be proved using the detailed balance condition [13] and considering the extended target pdf $\pi_g(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N)$. Furthermore, for $N = 1$ it is possible to show that SMH becomes the standard MH method with an independent proposal pdf.

5.2. Benefit of the interaction via MCMC

The use of the MCMC step facilitates the movements of the means toward the region of high probabilities of the target, independently of the choice of the initial parameters. Namely, it can help to reallocate "lost" means in a better position. This step could stop an explorative random walk behaviour of some proposal and reallocate it around a mode. It could be also applied to improve the initial positions of the means.

An important feature is that the impoverishment of the diversity in the population is reduced, compared to a standard resampling procedure [8, 9]. Moreover, the positions of the Gaussians will hardly ever change when the parameter of the SMH proposal φ are not properly chosen, since the new points will never be accepted. Thus, in the worst case, we would simply waste computational power by performing the MCMC operations. Another interesting point is that only one new importance weight needs to be evaluated at each iteration, since the rest of weights have already been computed in the previous steps (with the exception of the initial iteration, where all the weights need to be computed).

Finally, note that the parameters of the proposal $\varphi(\boldsymbol{\mu})$ for the SMH algorithm could be also adapted using different strategies already proposed in literature [16, 35]. Moreover, in our framework, the adaptation can be improved by using the built estimators $\hat{I}_t$ in APIS to update the parameters of the proposal (not only considering the samples generated on-line by the MCMC technique).

6. RELATIONSHIP WITH AMIS AND PMC AND CONSISTENCY OF THE ESTIMATORS

6.1. Estimators in PMC and AMIS: relationship with APIS

To clarify the different estimators used in PMC, AMIS and APIS, we distinguish two different phases of the sharing of the statistical information among the proposal pdfs:

- *In the space ($\mathbf{x} \in \mathcal{X}$):* creating an estimator sharing information among different proposal pdfs (forming a *population*) at the same time step.
- *In the time ($t \in \mathbb{N}$)*: combining estimators or information obtained in different iterations (creating a global estimator).

The PMC schemes use a cloud of proposal pdfs in each iterations (spread in the *space* of the variable of interest) and, in general, they use the standard IS approach to provide the estimator (see Appendix .1). The temporal combination of the information, i.e., the global estimator, can be built in different way, but the importance weights are in general based on the standard IS approach. In the numerical simulations, we also consider a *modified version* of PMC (M-PMC) where the spatial sharing of the information is done via deterministic mixture (see Appendix .2): this idea can be founded, jointly with the possibility of adapting a mixture, in the *Rao-Blackwellised version of the D-kernel PMC* algorithm [19, 20]. The way to update the mixture of proposals is much more complicated than in APIS, using the Kullback-Liebler divergence to tune the parameters. Furthermore, in [21], the authors suggest a procedure to adapt all the parameter of a mixture of pdfs. However, the resulting algorithm is sensitive to the initial conditions and, as a consequence, quite unstable.

The AMIS algorithm, at least in its basic formulation, uses and adapts only one proposal pdf at each iteration (N samples are drawn each step but from the same proposal). However, *all* the previous adapted proposal pdfs are considered to build a global estimator, following the determinist mixture approach. This is clearly the most stable way to construct the global estimator but quite costly since, each iteration, all the past proposal pdfs need to be evaluated.

APIS, in this sense, is "in between" PMC and AMIS: we use the deterministic mixture idea in the *space* at each iteration, as shown in Eqs. (4)-(5), whereas we use the standard IS approach to build the global estimator (in *time*), as shown in Eq. (6). Then, APIS is less costly than AMIS since APIS does not need the evaluation of the past proposal pdfs.

6.1.1. Consistency of the estimators

In all these schemes, PMC, AMIS and APIS, the consistency of the global estimator must be ensured when N (number of samples at each step) and t (iteration of the algorithm) grow to infinity. For $N \to \infty$, at a fixed iteration t, it is apparent using

Table 2. Comparison among the AIS, PMC, AMIS and APIS algorithms.

Algorithm	**Approaches in the estimation**		**Adaptation**		
	Space	**Time**	**type of IS weights**	**Memory**	**Equilibrium**
Standard Adaptive IS	none (single proposal)	Standard IS	Standard IS	Long	Static
(Basic) Generic PMC [9, 17]	Standard IS	Standard IS	Standard IS (Resampling)	Short (Resampling done on the current cloud)	Dynamic
Modified PMC (similarly in [19, 20])	Deterministic mixture	Standard IS	Standard IS (Resampling)	Short (Resampling done on the current cloud)	Dynamic
AMIS [18]	none (single proposal)	Deterministic mixture	Deterministic mixture	Long	Static
Modified AMIS [38]	none (single proposal)	Deterministic mixture	Standard IS	Short (like epochs in APIS)	Both Dynamic/ Pseudo-static (with N fixed)
APIS	Deterministic mixture	Standard IS	Standard IS	Short (epochs)	Both Dynamic/ Pseudo-static
Markov APIS	Deterministic mixture	Standard IS	Standard IS (+ MCMC)	Short (epochs)	Dynamic

the standard IS arguments [9]. For $t \to \infty$ and N finite, the situation is more complex, even if the standard IS approach (the simplest one) is used. Indeed, in this case (looking Eq. (6) and (19)), we have a convex combination of different independent[3] IS estimators that are all consistent, with $N \to \infty$, but *biased* [9]. Indeed, it is well-known that this bias is created by indirect estimation of the normalizing constant of π, i.e., Z, that we replace and estimate using with the sum of the weights. For $t \to \infty$, the variance of the global estimator vanishes to zero (since we are incorporating always new statistical information, provided by new independent samples) but the bias could be non-zero. However, the estimation of the normalizing constant of π converges quickly to the true value for $t \to \infty$, so that "unbiasedness property is still approximately held", as stated and discussed in [9, Chapter 14]), for PMC schemes. Namely, the bias vanishes to zero when t grows, i.e., the unbiasedness property "is approximately recovered" [9, Chapter 14]). This scenario applies to APIS as well, since APIS uses also the standard IS approach for building the global estimator. From another point of view, we can observe that Eq. (6) builds an increasingly (with t) valid standard IS estimator, as explained in Appendix .1 (considering the complete proposal in Eq. (12) at each time t).

In AMIS, the analysis for $t \to \infty$ is more complicated [38]. Indeed, the AMIS algorithm introduces a long memory dependence between the samples, due to the use of the deterministic mixture *in time*. Moreover, in AMIS, the IS weights (built using the deterministic mixture) are also used to adapt the proposal pdf, so that the long memory dependence yields a bias that can not be easily controlled by theoretical results. A similar and well-known problem appears with adaptive MCMC techniques: even if the kernel of the algorithm is valid at each step, a wrong adaptation (change of the kernel), using the all past generated samples, can jeopardize the convergence of the chain. Thus, to prove the consistency of AMIS, the authors in [38] suggests a modification in the adaptive structure of AMIS. In this modificated technique, the adaptation is given only considering the last generated samples (in the "APIS slang": in one epoch) and only standard IS weights, whereas the global estimation still uses the determinist mixture approach. Note that this resembles the adaptive structure of APIS. This reinforces the idea of APIS as robust technique, due to its memoryless feature in the adaptation. Table 2 summarizes the different features of PMC, AMIS and APIS schemes.

[3]The positions of the proposals depend on the previous configuration of the cloud but, the samples generated at each iteration are drawn independently from the previous ones and independent to each other, producing different independent IS estimators.

6.1.2. Evolution of proposals

In AMIS, the parameters of the proposal are updated and they converge to fixed values after a certain adequate number of iterations (as in a standard adaptive IS). Thus, the differences between two proposals at different time steps diminishes when $t \to \infty$. In the basic PMC schemes, the location parameters of the cloud of proposals are updated via resampling. In this case, the positions of the proposals change in each iteration, moving around the modes of the target (as a "dynamic equilibrium"). In APIS, similar situations can occur as shown in Fig. 1. Indeed, random walks around high probability regions can be generated but, unlike in PMC, in this case due to partial memoryless IS estimations or MCMC iterations (instead of the resampling procedure). Nevertheless, some proposal could also reach a pseudo-static equilibrium as in AMIS, for instance becoming trapped in a local mode. Both behaviours present certain advantages: APIS benefits of both features, aiming for a trade off between explorative search and stability in the estimation.

7. TOY EXAMPLE: NUMERICAL COMPARISON

7.1. Target distribution

In order to test and compare APIS with different other algorithms, we first consider a bivariate multimodal target pdf, which is itself a mixture of 5 Gaussians, i.e.,

$$\pi(\mathbf{x}) = \frac{1}{5}\sum_{i=1}^{5} \mathcal{N}(\mathbf{x}; \nu_i, \mathbf{\Sigma}_i), \quad \mathbf{x} \in \mathbb{R}^2, \tag{13}$$

with means $\nu_1 = [-10, -10]^\top$, $\nu_2 = [0, 16]^\top$, $\nu_3 = [13, 8]^\top$, $\nu_4 = [-9, 7]^\top$, $\nu_5 = [14, -14]^\top$, and covariance matrices $\mathbf{\Sigma}_1 = [2,\ 0.6; 0.6,\ 1]$, $\mathbf{\Sigma}_2 = [2,\ -0.4; -0.4,\ 2]$, $\mathbf{\Sigma}_3 = [2,\ 0.8; 0.8,\ 2]$, $\mathbf{\Sigma}_4 = [3,\ 0; 0,\ 0.5]$ and $\mathbf{\Sigma}_5 = [2,\ -0.1; -0.1,\ 2]$. Fig. 1 shows a contour plot of $\pi(\mathbf{x})$. Clearly, we can compute analytically moments of the target in Eq. (13), so that we can easily check the performance of the different techniques.

7.2. Goal, comparisons and initialization

We consider the problem of compute (a) the mean of the target, i.e., $E[\mathbf{X}] = [1.6, 1.4]^\top$ where $\mathbf{X} \sim \frac{1}{Z}\pi(\mathbf{x})$, (b) and the normalizing constant $Z = 1$, using Monte Carlo techniques. We compare the performance in term of Mean Square Error (MSE) in the estimation using different sampling methodologies: **(1)** standard, non-adaptive, Multiple IS (MIS) approach; **(2)** PIS (or static APIS) scheme; **(3-4)** APIS and MAPIS (APIS with the MCMC interaction) methods; **(5)** the AMIS technique [18]; **(6)** and a PMC acheme [17]. Moreover, we consider for all the previous techniques the two different initializations:

In1: First, we choose deliberately a "bad" initialization of the initial means to test the robustness of the algorithms and their ability to improve the corresponding *static* approaches. Specifically, the initial location parameters are selected uniformly within a square,

$$\boldsymbol{\mu}_i^{(0)} \sim \mathcal{U}([-4, 4] \times [-4, 4]),$$

for $i = 1, \ldots, N$. A single realization of $\boldsymbol{\mu}_i^{(0)}$ is depicted by the squares in Fig. 1(a) (jointly with the final locations $\boldsymbol{\mu}_i^{(T)}$, in one specific run).

In2: We also consider a better initialization. Specifically, the initial means are selected uniformly within a square,

$$\boldsymbol{\mu}_i^{(0)} \sim \mathcal{U}([-20, 20] \times [-20, 20]),$$

for $i = 1, \ldots, N$. A single realization of $\boldsymbol{\mu}_i^{(0)}$ is depicted by the squares in Fig. 1(b) (jointly with the final locations $\boldsymbol{\mu}_i^{(T)}$, in one specific run).

Below we provide more details of each applied scheme (providing the used parameters).

7.3. Techniques

We apply the following techniques proposed in this paper:
• *APIS:* we apply APIS with $N = 100$ Gaussian proposals

$$q_i^{(m)}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^{(m)}, \mathbf{C}_i), \qquad i = 1, \ldots, N.$$

The initial configurations of the means $\boldsymbol{\mu}_i^{(0)}$ are described above. First, we use the same isotropic covariance matrix, $\mathbf{C}_i = \sigma^2\mathbf{I}_2$, for every proposal. We test different values of $\sigma \in \{0.5, 1, 2, 3, 5, 7, 10, 20, 70\}$, to gauge the performance of APIS. Then, we also try different non-isotropic diagonal covariance matrices, $\mathbf{C}_i = \text{diag}(\sigma_{i,1}^2, \sigma_{i,2}^2)$, where $\sigma_{i,j} \sim \mathcal{U}([1, 10])$ for $j \in \{1, 2\}$ and $i = 1, \dots N$, i.e., different for each proposal. We set $T = 2000$ and $T_a \in \{2, 5, 20, 50, 100\}$, i.e., $M = \frac{T}{T_a} \in \{20, 40, 100, 400, \frac{T}{2} = 1000\}$. We test the performance of APIS with the two inizializatons described above **In1** and **In2**.

• *PIS (static APIS)*: we also consider the case $M = 1$, which corresponds to a static APIS technique with multiple proposals and no adaptation. PIS combines the deterministic mixture idea and the standard IS approach to build the global estimators (see Appendix 9). For this reason, is different from a standard multiple IS scheme, see below.

• *MAPIS*: for the MCMC interaction, we consider again a Gaussian proposal for the SMH method, i.e., $\varphi(\boldsymbol{\mu}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \lambda^2\mathbf{I}_2)$, with $\lambda = 10$. To maintain a constant computational cost in each simulation, we fix $\Upsilon = T_a = \frac{T}{M}$ (the number of iterations of SMH, at the end of each epoch), i.e., the total number of iterations of SMH in the entire MAPIS method is alway $M\Upsilon = T$.

Moreover, we compare these techniques with the following benchmark schemes:

• *Non-adaptive Multiple IS (MIS):* Given the initial $\boldsymbol{\mu}_i^{(0)}$, these positions never change as in PIS. We set $N = 100$. Thus, $T = 2000$ samples are drawn from each proposal in order to a fair comparison with APIS (since in APIS we use $NT = 20000$ samples). The IS weights are built using the standard IS approach described in Appendix .1.

• *AMIS scheme:* AMIS uses only one proposal pdf in the space fixing the temporal iteration index m, i.e.,

$$h_m(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \boldsymbol{\Phi}_m), \quad m = 0, \dots, M-1,$$

Both parameters $\boldsymbol{\mu}_m$ and $\boldsymbol{\Phi}_m$ are updated each iteration. Note that we have used M as the number of adaptive iterations in AMIS since it is equivalent to the number of epochs M used in APIS. The initial mean $\boldsymbol{\mu}_0$ is chosen according **In1** and **In2**, whereas $\boldsymbol{\Phi}_0 = \sigma^2\mathbf{I}_2$ with $\sigma \in \{0.5, 1, 2, 3, 5, 7, 10, 20, 70\}$. At each iteration m, K samples are drawn from $h_m(\mathbf{x})$. Then, IS weights are associated to these samples using the deterministic mixture idea, taking in account all the previous proposals $h_0(\mathbf{x})$, $h_1(\mathbf{x})$,..., $h_{m-1}(\mathbf{x})$. Therefore, the weights associated to previous samples need to be updated as well. For these reasons, AMIS is more costly than APIS, in general. Then, the parameters are updated $\boldsymbol{\mu}_m$ and $\boldsymbol{\Phi}_m$ according to the IS estimation of the mean and variances of the target. We have consider a pairs of K and M such that $KM = NT = 20000$, for a fair comparison with APIS. Specifically, we have run different simulations using $K = \{500, 1000, 2000, 5000\}$ and, as a consequence, $M = \{40, 20, 10, 4\}$. Obviously, AMIS becomes more costly when M increases. However, depending on the stating value σ, the best results of AMIS in this scenario are usually provided by $M = 4, 10$ (i.e., $K = 5000$ and $K = 2000$). This is due to better estimations of the mean and covariance of the target is achieved, so that the adaptation is also improved.

• *PMC schemes:* we also apply the mixture PMC scheme [17]. More precisely, we consider a population of samples

$$\{\mathbf{x}_1^{(t)}, \dots, \mathbf{x}_N^{(t)}\},$$

at the t-th iteration, and propagate them with random walks

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \epsilon_t, \quad i = 1, \dots, N,$$

where $\epsilon_t \sim \mathcal{N}(\mathbf{x}; [0, 0]^\top, \boldsymbol{\Phi})$, with $\boldsymbol{\Phi} = \sigma^2\mathbf{I}_2$ and $\sigma \in \{0.5, 1, 2, 5, 10, 20, 70\}$. At each iteration, the resampling step is performed according to the normalized importance weights. The initial cloud $\{x_i^{(1)}\}_{i=1}^N$ is chosen according with the same initialization procedure described above **In1** and **In2**. The cumulative mean of the cloud $\{x_i^{(t)}\}_{i=1,t=1}^{N,T}$, as well as the cumulative estimate of the normalizing constant, are computed until $T = 2000$. We have not been able to apply the adaptive strategy suggested in [17] in order to select suitable scale parameters, within a population of pre-chosen values, since it has been difficult to select these values adequately. More specifically, we have not been able to find a set of parameters for this approach that provides reasonable results in this scenario. We have set $N = 100$ for a fair comparison with APIS, using the same total number of samples NT. Moreover, we have also run other simulations with $N = 500, 2000$ in order to see the computational cost needed to reach the APIS performance. Finally, we have also considered a Modified PMC (M-PMC) that, similarly in [19, 20], uses the deterministic mixture for the spatial construction of the global estimator as in APIS. The results are shown in Tables 3-4.

7.4. Results

All the results are averaged over 2000 independent experiments. Tables 3 and 4 show the Mean Square Error (MSE) in estimation of the mean (first component), with the initialization **In1** and **In2**, for the different algorithms. In AMIS, for sake of simplicity, we directly show the worst and best results among the several simulations made with different parameters (see

a detailed description above). The results of MAPIS and PMC with $N = 500, 2000$ are included in two different subtables since their application entails more computational effort. In each subtables, the best results in each column are highlighted in bold-face.

We can observe that APIS outperforms the other techniques, except for a few values of σ, where APIS has a negligibly larger error. Only with $\sigma = 70$, AMIS has an MSE sensibly smaller than APIS in its best case. However, this result depends strictly on the choice of the parameter: the MSE of AMIS in its worst case is the highest whereas APIS provides always small MSE independently from the choice of T_a. Moreover, for high values of $\sigma \in \{10, 20, 70\}$, the APIS results could be easily improved using a higher value of T_a (for instance, $T_a = 500$). Observe also that the robust implementation, choosing randomly the scale parameters $\sigma_{i,j} \sim \mathcal{U}([1, 10])$, provide the best results (with the exception of PMC with $N = 2000$ which provides negligibly smaller MSE, with much higher computational cost). Moreover, MAPIS in general improves the results and the robustness of APIS, but being more costly due to the additional MCMC steps. Figures 2 depict the MSE in log-scale of the estimation of mean of π versus the choice of the scale parameters $\sigma_{i,j}$, comparing the different techniques.

Table 3.1

Alg. \ Std		$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 5$	$\sigma = 7$	$\sigma = 10$	$\sigma = 20$	$\sigma = 70$	$\sigma_{i,j} \sim \mathcal{U}([1, 10])$
MIS		29.56	41.95	64.51	42.84	2.17	0.0454	0.0147	0.0187	0.1914	4.55
PIS ($T_a = T$)		29.28	47.74	75.22	17.61	0.2424	0.0280	0.0124	0.0176	0.1789	0.0651
APIS	$T_a = 100$	22.86	13.70	6.2606	2.47	0.0438	0.0131	0.0129	0.0212	0.1821	0.0110
	$T_a = 50$	17.62	12.14	5.42	1.99	0.0501	0.0118	0.0138	0.0209	0.1750	0.0077
	$T_a = 20$	14.75	11.33	4.77	1.66	0.0361	**0.0108**	0.0146	0.0208	0.1873	0.0056
	$T_a = 5$	13.01	8.50	2.30	0.2831	**0.0074**	0.0114	0.0149	0.0251	0.2027	**0.0045**
	$T_a = 2$	**9.46**	**2.45**	**0.0225**	**0.0170**	0.0103	0.0139	0.0185	0.0354	0.2007	0.0077
AMIS	(best)	124.22	121.21	100.23	54.67	0.8640	0.0124	**0.0121**	**0.0126**	**0.0136**	– – – – –
	(worst)	125.43	123.38	114.82	89.09	16.92	0.3626	0.0128	0.0131	18.66	– – – – –
PMC	$N = 100$	112.99	114.11	47.97	26.32	2.34	0.5217	0.0559	0.4331	2.41	0.3017
M-PMC	$N = 100$	111.92	107.58	26.86	6.03	0.6731	0.1154	0.0744	0.4142	2.42	0.07

Table 3.2

MAPIS ($N = 100$)	$T_a = 100$	0.7134	0.0933	0.3213	0.1611	0.0167	0.0101	0.0147	0.0023	0.1765	0.0070
	$T_a = 50$	0.7058	0.1287	0.1136	0.1097	0.0114	**0.0094**	0.0139	0.0020	0.1831	0.0051
	$T_a = 20$	0.6950	0.1319	0.0464	0.1040	**0.0081**	0.0098	0.0152	0.0021	0.1943	**0.0041**
	$T_a = 5$	0.2729	0.0665	0.0319	0.0154	0.0082	0.0123	0.0151	0.0019	0.1946	0.0046
	$T_a = 2$	**0.1708**	**0.0148**	**0.0116**	**0.0138**	0.0105	0.0130	0.0165	0.0027	0.1918	0.0075
PMC	$N = 500$	112.18	113.10	36.63	18.59	2.20	0.4011	0.0134	0.0259	0.8891	0.2964
	$N = 2000$	112.09	112.45	27.91	13.63	2.01	0.1899	**0.0057**	**0.0028**	**0.1120**	0.2802

Table 3. MSE of the estimation of the mean of the target (first component) with the initialization **In1**, using MIS, PIS, APIS, MAPIS, AMIS and PMC. In Table 3.1, we set $N = 100$ ($T = 2000$; the total number of samples is $NT = 20000$) for MIS, PIS, APIS and PMC. Recall that, with AMIS, we have one proposal each iteration. Moreover, for AMIS, we show directly the best results, obtained varying K and M such that $KM = NT = 20000$ (as detailed in the text). In Table 3.2, the MSE of MAPIS and PMC with $N = 500, 2000$, is given. We have indicated the best results in each sub-table with bold-faces.

8. LOCALIZATION PROBLEM IN A WIRELESS SENSOR NETWORK

We consider the problem of positioning a target in a 2-dimensional space using range measurements. This is a problem that appears frequently in localization applications in wireless sensor networks [32, 33, 34]. Namely, we consider a random vector $\mathbf{X} = [X_1, X_2]^\top$ to denote the target position in the plane $\mathbb{R}^2$. The position of the target is then a specific realization $\mathbf{X} = \mathbf{x}$. The range measurements are obtained from 3 sensors located at $\mathbf{h}_1 = [-10, 2]^\top$, $\mathbf{h}_2 = [8, 8]^\top$ and $\mathbf{h}_3 = [-20, -18]^\top$. The observation equations are given by

$$Y_j = a \log\left(\frac{||\mathbf{x} - \mathbf{h}_j||}{0.3}\right) + \Theta_j, \quad j = 1, \ldots, 3, \tag{14}$$

where Θ_j are independent Gaussian variables with identical pdfs, $\mathcal{N}(\vartheta_j; 0, \omega^2)$, $j = 1, 2$. We also consider a prior density over ω, i.e., $\Omega \sim p(\omega) = \mathcal{N}(\omega; 0, 25)I(\omega > 0)$, where $I(\omega > 0)$ is 1 if $\omega > 0$ and 0 otherwise. The parameter $A = a$ is also

Table 4.1

Alg. \ Std		$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 5$	$\sigma = 7$	$\sigma = 10$	$\sigma = 20$	$\sigma = 70$	$\sigma_{i,j} \sim \mathcal{U}([1, 10])$
MIS		12.00	9.40	10.26	10.64	7.67	4.40	0.5443	0.0321	0.1764	4.37
PIS ($T_a = T$)		10.14	0.9469	0.0139	0.0085	0.0100	0.0115	0.0146	0.0237	0.1756	0.0106
APIS	$T_a = 100$	0.7741	0.0318	0.0011	0.0017	0.0054	0.0118	**0.0129**	0.0211	0.1794	0.0032
	$T_a = 50$	0.5792	0.0144	0.0007	0.0015	0.0051	0.0112	0.0131	0.0221	0.1772	**0.0029**
	$T_a = 20$	0.4831	0.0401	0.0006	**0.0014**	**0.0047**	**0.0095**	0.0136	0.0245	0.1732	**0.0029**
	$T_a = 5$	0.2552	**0.0008**	**0.0005**	0.0022	0.0064	0.0111	0.0149	0.0270	0.2076	0.0039
	$T_a = 2$	**0.0547**	0.0017	0.0116	0.0051	0.0103	0.0142	0.0182	0.0387	0.1844	0.0080
AMIS	(best)	113.97	112.70	107.85	91.56	44.93	12.75	0.7404	**0.0121**	**0.0141**	$- - - - -$
	(worst)	116.66	115.62	111.83	104.44	70.62	35.66	9.43	0.0871	18.62	$- - - - -$
PMC	$N = 100$	111.54	110.78	90.21	46.84	2.29	0.5023	0.0631	0.4273	2.42	0.3082
M-PMC	$N = 100$	23.16	7.43	7.56	3.11	0.6420	0.1173	0.0720	0.4194	2.37	0.0695

Table 4.2

MAPIS ($N = 100$)	$T_a = 100$	0.4753	0.0334	0.0027	0.0017	**0.0059**	0.0092	0.0135	0.0217	0.1762	0.0034
	$T_a = 50$	0.4677	0.0287	0.0007	0.0015	**0.0059**	**0.0091**	0.0133	0.0222	0.1901	0.0031
	$T_a = 20$	0.3110	0.0092	**0.0006**	**0.0014**	0.0061	**0.0091**	0.0141	0.0233	0.1805	**0.0030**
	$T_a = 5$	0.3497	0.0015	0.0007	0.0041	0.0079	0.0122	0.0155	0.0249	0.1933	0.0039
	$T_a = 2$	**0.0870**	**0.0101**	0.0028	0.0060	0.0098	0.0126	0.0154	0.0333	0.2026	0.0078
PMC	$N = 500$	110.58	109.69	64.81	15.99	2.09	0.4841	0.0144	0.0267	0.8924	0.2900
	$N = 2000$	108.22	107.10	27.93	13.21	1.84	0.1912	**0.0054**	**0.0027**	**0.0988**	0.2805

Table 4. MSE of the estimation of the mean of the target (first component) with the initialization **In2**, using MIS, PIS, APIS, MAPIS, AMIS and PMC. In Table 4.1, we set $N = 100$ ($T = 2000$; the total number of samples is $NT = 20000$) for MIS, PIS, APIS and PMC. Recall that, with AMIS, we have one proposal each iteration. Moreover, for AMIS, we show directly the best results, obtained varying K and M such that $KM = NT = 20000$ (as detailed in the text). In Table 4.2, the MSE of MAPIS and PMC with $N = 500, 2000$, is given. We have indicated the best results in each sub-table with bold-faces.

unknown and we again consider a Gaussian prior $A \sim p(a) = \mathcal{N}(a; 0, 25)$. Moreover, we also apply Gaussian priors over $\mathbf{X}$, i.e., $p(x_i) = \mathcal{N}(x_i; 0, 25)$ with $i = 1, 2$. Thus, the posterior pdf $\pi(x_1, x_2, a, \omega) = p(x_1, x_2, a, \omega|\mathbf{y})$ is

$$\pi(x_1, x_2, a, \omega) \propto p(\mathbf{y}|x_1, x_2, a, \omega)p(x_1)p(x_2)p(a)p(\omega),$$

where $\mathbf{y} \in \mathbb{R}^d$ is the vector of received measurements. We simulate $d = 30$ observations from the model ($d/3 = 10$ from each of the three sensors) fixing $x_1 = 3$, $x_2 = 3$, $a = -20$ and $\omega = 5$. With $d = 30$, the expected value of the target ($E[X_1] \approx 2.8749$, $E[X_2] \approx 3.0266$, $E[A] \approx 5.2344$, $E[\Omega] \approx 20.1582$)[4] is quite close to the true values.

Our goal is computing the expected value of $(X_1, X_2, A, \Omega) \sim \pi(x_1, x_2, a, \omega)$ via Monte Carlo, in order to provide an estimation of the position of the target, the parameter a and the standard deviation ω of the noise in the system. We apply APIS and PMC schemes both using N Gaussian proposals as in the previous example. For both algorithm, we initialize the cloud of particles spread out in the space of the variables of interest, i.e.,

$$\boldsymbol{\mu}_i^{(0)} \sim \mathcal{N}(\boldsymbol{\mu}; \mathbf{0}, 30^2\mathbf{I}_4), \quad i = 1, ..., N,$$

and the scale parameters $\mathbf{C}_i = \sigma_{i,j}^2\mathbf{I}_4$ with $j = 1, \ldots, 4$ and $i = 1, \ldots, N$. The values of the standard deviations $\sigma_{i,j}$ are chosen randomly for each Gaussian pdf. Specifically, $\sigma_{i,j} \sim \mathcal{U}([1, Q])$, where we have considered three possible values for Q, i.e., $Q \in \{5, 10, 30\}$.

The MSE of the estimation (averaged over 3000 independent runs) are provided in Table 5 and 6 for different values of $N \in \{50, 100, 200\}$, $T \in \{1000, 2000, 4000\}$ and $T_a \in \{20, 100\}$. More specifically in Table 5, we maintain fixed $T = 2000$ whereas, in Table 6 we keep fixed the total number of generated samples $NT = 2\ 10^5$. APIS outperforms always PMC when $\sigma_{i,j} \sim \mathcal{U}([1, 5])$ and $\sigma_{i,j} \sim \mathcal{U}([1, 10])$ whereas PMC provides better results for $\sigma_{i,j} \sim \mathcal{U}([1, 30])$ (with the exception of the case $N = 200$ and $T = 2000$ in Table 5). This is owing to APIS, in this case with bigger variances, needs the use of a greater value of T_a. Therefore, the results show jointly the robustness and flexibility of the APIS technique.

[4]These values have been obtained with a deterministic, expensive and exhaustive numerical integration method, using a thin grid.

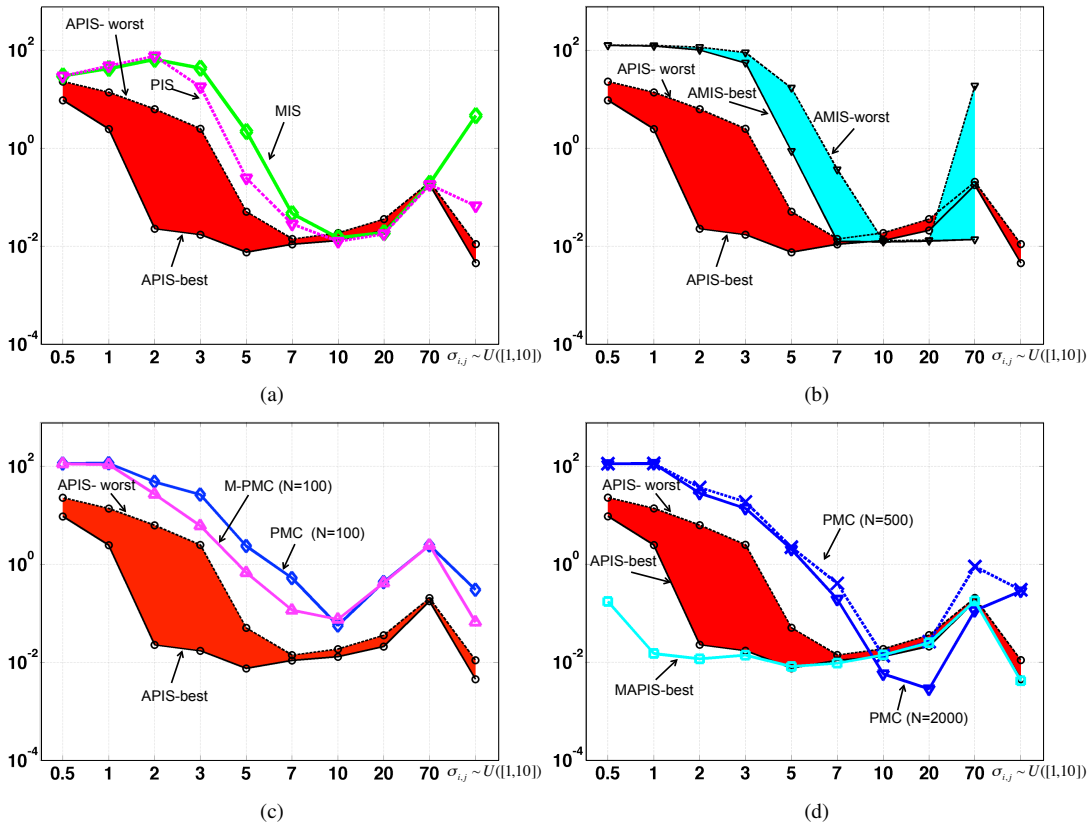

Fig. 2. MSE in log-scale versus the scale parameters, $\sigma_{i,j}$, in the estimation of the first component of the expected value of π. For the APIS and AMIS methods, we show the best and worst results. **(a)** Comparison among MIS (rhombus), PIS (triangles) and APIS. **(b)** Comparison between AMIS and APIS. **(c)** Comparison among M-PMC (rhombus), PMC with $N = 100$ (triangles) and APIS. **(d)** Comparison among PMC with $N = 500$ (X-marks), PMC with $N = 2000$ (triangles), APIS and MAPIS (squares).

9. CONCLUSIONS

In this work, we have introduced the adaptive population importance sampling (APIS) algorithm. APIS is an iterative importance sampling (IS) technique which uses multiple adaptive proposal pdfs. On the one hand, the deterministic mixture is used to build the partial IS estimators for the population of proposals in APIS, thus providing an increased robustness w.r.t. the population Monte Carlo (PMC) approach. On the other hand, the temporal evolution makes use of a standard IS estimator, thus avoiding the increase in computational cost as the algorithm evolves occurring in the adaptive multiple importance sampling (AMIS) scheme. Consequently, APIS is able to attain simultaneously the advantages of these two approaches (simplicity and robustness) while minimizing their drawbacks. Differently from PMC, APIS updates the proposal pdfs in an adaptive IS fashion, without using resampling. Hence, since no resampling is performed, there is no loss of diversity in the mixture of proposals. Furthermore, by introducing an MCMC approach on top of APIS (thus leading to the so-called MAPIS algorithm) that diversity may be increased w.r.t. the initial population. Numerical results confirm that APIS outperforms both techniques (AMIS and PMC) in terms of performance and robustness w.r.t. the choice of the initial parameters.

Acknowledgment

This work has been supported by the Spanish government's projects COMONSENS (CSD2008-00010), ALCIT (TEC2012-38800-C03-01), DISSECT (TEC2012-38058-C03-01), OTOSiS (TEC2013-41718-R), and COMPREHENSION (TEC2012-38883-C02-01), by the ERC grant 239784 and AoF grant 251170, and by the European Union 7th Framework Programme through the Marie Curie Initial Training Network "Machine Learning for Personalized Medicine" MLPM2012, Grant No.

Alg. \ Std			$\sigma_{i,j} \sim \mathcal{U}([1,5])$	$\sigma_{i,j} \sim \mathcal{U}([1,10])$	$\sigma_{i,j} \sim \mathcal{U}([1,30])$
APIS	$N = 50$	$T_a = 20$	**0.0181**	**0.0768**	0.8298
	$N = 50$	$T_a = 100$	0.0261	0.0770	0.6808
PMC	$N = 50$	—	0.2067	0.5224	**0.2421**
APIS	$N = 100$	$T_a = 20$	**0.0073**	**0.0379**	0.5075
	$N = 100$	$T_a = 100$	0.0147	0.0433	0.3702
PMC	$N = 100$	—	0.0642	0.4345	**0.1533**
APIS	$N = 200$	$T_a = 20$	**0.0053**	**0.0174**	0.2816
	$N = 200$	$T_a = 100$	0.0111	0.0229	**0.1886**
PMC	$N = 200$	—	0.0136	0.2741	0.3455

Table 5. MSE of the estimation of $E[(X_1, X_2, A, \Omega)]$ using APIS and PMC with $T = 2000$, for different random choices of the scale parameters and different number of particles N in the population. The best results, in each column and with the same number N, are highlighted with bold-faces.

Alg. \ Std				$\sigma_{i,j} \sim \mathcal{U}([1,5])$	$\sigma_{i,j} \sim \mathcal{U}([1,10])$	$\sigma_{i,j} \sim \mathcal{U}([1,30])$
APIS	$N = 50$	$T = 4000$	$T_a = 20$	**0.0067**	0.0330	0.4827
	$N = 50$	$T = 4000$	$T_a = 100$	0.0069	**0.0269**	0.3078
PMC	$N = 50$	$T = 4000$	—	0.2014	0.5151	**0.2214**
APIS	$N = 200$	$T = 1000$	$T_a = 20$	**0.0139**	**0.0498**	0.5800
	$N = 200$	$T = 1000$	$T_a = 100$	0.0453	0.0947	0.5439
PMC	$N = 200$	$T = 1000$	—	0.0153	0.2747	**0.3540**

Table 6. MSE of the estimation of $E[(X_1, X_2, A, \Omega)]$ using APIS and PMC, for different random choices of the scale parameters, keeping constant the total number of drawn samples $NT = 2\ 10^5$. The best results, in each column and with the same number N, are highlighted with bold-faces.

316861.

10. REFERENCES

[1] P. M. Djurić and S. J. Godsill, Eds., *Special Issue on Monte Carlo Methods for Statistical Signal Processing*, IEEE Transactions Signal Processing 50 (3), February 2002.

[2] P. M. Djurić, J. Zhang, T. Ghirmai, Y. Huang, and J. H. Kotecha, "Applications of particle filtering to communications: A review," in *Proceedings of the XI EUSIPCO*. EURASIP, September 2002.

[3] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, September 2003.

[4] X. Wang, R. Chen, and J. S. Liu, "Monte Carlo Bayesian signal processing for wireless communications," *Journal of VLSI Signal Processing*, vol. 30, pp. 89–105, 2002.

[5] A. Doucet and X. Wang, "Monte Carlo methods for signal processing," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 152–170, Nov. 2005.

[6] W. J. Fitzgerald, "Markov chain Monte Carlo methods with applications to signal processing," *Signal Processing*, vol. 81, no. 1, pp. 3–18, January 2001.

[7] J. Kotecha and P. M. Djurić, "Gaussian particle filtering," *IEEE Transactions Signal Processing*, vol. 51, no. 10, pp. 2592–2601, October 2003.

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

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

[10] N. Chopin and C. P. Robert, "Properties of nested sampling," *Biometrika*, vol. 97, no. 3, pp. 741–755, 2010.

[11] J. Skilling, "Nested sampling," *American Institute of Physics Conference Series*, vol. 119, pp. 1211–1232, 2004.

[12] J. Skilling, "Nested sampling for general Bayesian computation," *Bayesian Analysis*, vol. 1, no. 4, pp. 833–860, June 2006.

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

[14] J. Kotecha and P. M. Djurić, "Gibbs sampling approach for generation of truncated multivariate gaussian random variables," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999.

[15] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan, "An introduction to MCMC for machine learning," *Machine Learning*, vol. 50, pp. 5–43, 2003.

[16] 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.

[17] O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert, "Population Monte Carlo," *Journal of Computational and Graphical Statistics*, vol. 13, no. 4, pp. 907–929, 2004.

[18] J. M. Cornuet, J. M. Marin, A. Mira, and C. P. Robert, "Adaptive multiple importance sampling," *Scandinavian Journal of Statistics*, vol. 39, no. 4, pp. 798–812, December 2012.

[19] G.R. Douc, J.M. Marin, and C. Robert, "Convergence of adaptive mixtures of importance sampling schemes," *Annals of Statistics*, vol. 35, pp. 420–448, 2007.

[20] G.R. Douc, J.M. Marin, and C. Robert, "Minimum variance importance sampling via population Monte Carlo," *ESAIM: Probability and Statistics*, vol. 11, pp. 427–447, 2007.

[21] O. Cappé, R. Douc, A. Guillin, J. M. Marin, and C. P. Robert, "Adaptive importance sampling in general mixture classes," *Statistics and Computing*, vol. 18, pp. 447–459, 2008.

[22] E. Koblents and J. Miguez, "Robust mixture populationmonte carlo scheme with adaptation of the number of components," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*. IEEE, 2013, pp. 1–5.

[23] M. Bugallo, M. Hong, and P. M. Djuric, "Marginalized population Monte Carlo," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 2925–2928.

[24] M. Hong, M. F Bugallo, and P. M Djuric, "Joint model selection and parameter estimation by population monte carlo simulation," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 3, pp. 526–539, 2010.

[25] A. Owen and Y. Zhou, "Safe and effective importance sampling," *Journal of the American Statistical Association*, vol. 95, no. 449, pp. 135–143, 2000.

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

[27] L. Pozzi and A. Mira, "A R adaptive multiple importance sampling (ARAMIS)," *http://cran.r-project.org/web/packages/ARAMIS/vignettes/demoARAMIS.pdf*, 2012.

[28] J. Sirén, P. Marttinen, and J. Corander, "Reconstructing population histories from single nucleotide polymorphism data," *Mol Biol Evol*, vol. 28, no. 1, pp. 673–683, 2011.

[29] L. Martino, V. Elvira, D. Luengo, and J. Corander, "An adaptive population importance sampler," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 8088–8092, 2014.

[30] C. Berzuini and W. Gilks, "Resample-move filtering with cross-model jumps," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., chapter 6. Springer, 2001.

[31] E. Koblents and J. Miguez, "A population Monte Carlo scheme with transformed weights and its application to stochastic kinetic models," *Statistics and Computing*, 2014.

[32] A. M. Ali, K. Yao, T. C. Collier, E. Taylor, D. Blumstein, and L. Girod, "An empirical study of collaborative acoustic source localization," *Proc. Information Processing in Sensor Networks (IPSN07), Boston*, April 2007.

[33] 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*, vol. 23, no. 4, pp. 809–819, April 2005.

[34] L. Martino and J. Míguez, "Generalized rejection sampling schemes and applications in signal processing," *Signal Processing*, vol. 90, no. 11, pp. 2981–2995, November 2010.

[35] H. Haario, E. Saksman, and J. Tamminen, "An adaptive Metropolis algorithm," *Bernoulli*, vol. 7, no. 2, pp. 223–242, April 2001.

[36] T. Hesterberg, "Weighted average importance sampling and defensive mixture distributions," *Technometrics*, vol. 37, no. 2, pp. 185–194, 1995.

[37] E. Marinari and G. Parisi, "Simulated tempering: a new Monte Carlo scheme," *Europhysics Letters*, vol. 19, no. 6, pp. 451–458, July 1992.

[38] J. M. Marin, P. Pudlo, and M. Sedki, "Consistency of the adaptive multiple importance sampling," *arXiv:1211.2548*, 2012.

IS approaches using with multiple proposal densities

Recall that our goal is computing efficiently some moment of $\mathbf{x}$, i.e., an integral measure w.r.t. the target pdf $\frac{1}{Z}\pi(\mathbf{x})$, namely, $I = \frac{1}{Z}\int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$. If we consider to draw N_1 samples from one proposal pdf, i.e. $\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_{N_1}^{(1)} \sim q_1(\mathbf{x})$, and other N_2 from the other one, $\mathbf{x}_1^{(2)}, \ldots, \mathbf{x}_{N_2}^{(2)} \sim q_2(\mathbf{x})$ (q_1 and q_2 both normalized), there are at least two procedures to build a joint IS estimator, as shown in the following.

.1. Standard IS approach

The simplest approach [9, Chapter 14] is to compute the classical corresponding importance weights

$$w_i^{(1)} = \frac{\pi(\mathbf{x}_i^{(1)})}{q_1(\mathbf{x}_i^{(1)})}, \quad i = 1, \ldots, N_1; \tag{15}$$

$$w_j^{(2)} = \frac{\pi(\mathbf{x}_j^{(2)})}{q_2(\mathbf{x}_j^{(2)})}, \quad j = 1, \ldots, N_2. \tag{16}$$

The IS estimator $\hat{I}$ is built normalizing them jointly, i.e., computing $S_{tot} = \sum_{i=1}^{N_1} w_i^{(1)} + \sum_{j=1}^{N_2} w_j^{(2)}$ so that we can write

$$\hat{I} = \frac{1}{S_{tot}} \left(\sum_{i=1}^{N_1} w_i^{(1)} f(\mathbf{x}_i^{(1)}) + \sum_{j=1}^{N_2} w_j^{(2)} f(x_j^{(2)}) \right). \tag{17}$$

Note that, defining the partial sums $S_1 = \sum_{i=1}^{N_1} w_i^{(1)}$ and $S_2 = \sum_{j=1}^{N_2} w_j^{(2)}$ so that $S_{tot} = S_1 + S_2$ and considering the normalized weights,

$$\bar{w}_i^{(1)} = \frac{w_i^{(1)}}{S_1}, \qquad \bar{w}_j^{(2)} = \frac{w_j^{(2)}}{S_2},$$

Eq. (17) can be expressed as

$$\begin{aligned} \hat{I} &= \frac{1}{S_1 + S_2} \left(S_1 \sum_{i=1}^{N_1} \bar{w}_i^{(1)} f(\mathbf{x}_i^{(1)}) + S_2 \sum_{j=1}^{N_2} \bar{w}_j^{(2)} f(x_j^{(2)}) \right), \\ \hat{I} &= \frac{1}{S_1 + S_2} \left(S_1 \hat{I}_1 + S_2 \hat{I}_2 \right) = \frac{S_1}{S_1 + S_2} \hat{I}_1 + \frac{S_2}{S_1 + S_2} \hat{I}_2, \end{aligned} \tag{18}$$

where $\hat{I}_1, \hat{I}_2$ are the partial IS estimators considering only one proposal pdf. Clearly, this procedure can be extended for $K > 2$ different proposal pdfs, $k = 1, \ldots, K$, obtaining the complete estimator

$$\hat{I} = \frac{\sum_{k=1}^{K} S_k \hat{I}_k}{\sum_{k=1}^{N} S_k}, \tag{19}$$

as convex combination of the partial estimators $\hat{I}_k$ considering only the k-th proposal pdf.

.2. Deterministic mixture

Another approach is the so-called *determinist mixture* [25, 26]. Setting

$$\mathbf{z}_{1:N_1+N_2} = \left[\mathbf{x}^{(1)}_{1:N_1}, \mathbf{x}^{(2)}_{1:N_2}\right],$$

the weights are defined as

$$w_i = \frac{\pi(\mathbf{z}_i)}{q^*(\mathbf{z}_i)} = \frac{\pi(\mathbf{z}_i)}{\frac{N_1}{N_1+N_2} q_1(\mathbf{z}_i) + \frac{N_2}{N_1+N_2} q_2(\mathbf{z}_i)},$$

for $i = 1, \ldots, N_1 + N_2$. In this case, the *complete* proposal q^* is considered to be a *mixture* of q_1 and q_2 weighted according to the number of each corresponding samples. Here, we draw N_1 samples from q_1 and N_2 samples from q_2; these number are fixed and deterministic, unlike in the standard procedure of sampling from a mixture, where a pdf is randomly chosen according to its weight. However, the sample $\mathbf{z}'$drawn in this *deterministic* way, is exactly distributed as the mixture $q^*(\mathbf{z}') = \frac{N_1}{N_1+N_2} q_1(\mathbf{z}') + \frac{N_2}{N_1+N_2} q_2(\mathbf{z}')$. The IS estimator is then simply

$$\hat{I} = \frac{1}{\sum_{i=1}^{N_1+N_2} w_i} \sum_{i=1}^{N_1+N_2} w_i f(\mathbf{z}_i),$$

and for $K > 2$ proposal pdfs:

$$w_i = \frac{\pi(\mathbf{z}_i)}{\sum_{k=1}^{K} \left(\frac{N_k}{\sum_{k=1}^{K} N_k}\right) q_k(\mathbf{z}_k)}; \tag{20}$$

$$\hat{I} = \frac{1}{\sum_{i=1}^{N_1+\ldots+N_K} w_i} \sum_{i=1}^{N_1+\ldots+N_K} w_i f(\mathbf{z}_i). \tag{21}$$

In general, this IS estimator is more stable and provides better performance than the previous one [25]. On the other hand, it needs to evaluate each proposal in more points (all the generated samples, also drawn from the other proposals), and therefore it is more costly.

A. ESTIMATION OF THE PARTITION FUNCTION

Below we provide the MSE of the estimation of the normalizing constant of the target in Section 7. Table 7 is referred to the the initialization **I1** whereas Table 8 corresponds to the initialization **I2**.

Table 5.1

Alg. \ Std		$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 5$	$\sigma = 7$	$\sigma = 10$	$\sigma = 20$	$\sigma = 70$	$\sigma_{i,j} \sim \mathcal{U}([1, 10])$
MIS		$3.94 \cdot 10^4$	$7.12 \cdot 10^7$	$1.07 \cdot 10^3$	3.43	0.01	0.0003	0.0001	0.0001	0.0016	0.2190
PIS ($T_a = T$)		$9.51 \cdot 10^{12}$	$4.60 \cdot 10^5$	15.34	0.2523	0.0016	0.0002	0.0001	0.0002	0.0016	0.0005
APIS	$T_a = 100$	3.28	0.6252	0.0681	0.0179	0.0003	0.0001	0.0001	0.0002	0.0016	0.0001
	$T_a = 50$	1.37	0.4438	0.0489	0.0142	0.0004	0.0001	0.0001	0.0002	0.0017	0.0001
	$T_a = 20$	0.7717	0.3372	0.0375	0.0109	0.0002	0.0001	0.0001	0.0002	0.0017	0.0000
	$T_a = 5$	0.4996	0.1616	0.0154	0.0019	0.0001	0.0001	0.0001	0.0002	0.0018	0.0000
	$T_a = 2$	0.2127	0.0173	0.0002	0.0001	0.0001	0.0001	0.0001	0.0003	0.0018	0.0001
AMIS	(best)	15.92	15.66	12.81	4.49	0.0069	0.0001	0.0001	0.0001	0.0001	– – – – –
	(worst)	15.97	15.92	14.87	10.70	0.4559	0.0024	0.0001	0.0001	1.62	– – – – –
PMC	$N = 100$	33.53	17.10	14.42	5.06	0.4249	0.0553	0.0015	0.0003	0.0016	0.3542
M-PMC	$N = 100$	15.85	14.31	1.81	0.6546	0.0402	0.0021	0.0002	0.0002	0.0016	0.0004

Table 5.2

MAPIS	$T_a = 100$	0.7048	0.0933	0.3213	0.1611	0.0167	0.0101	0.0147	0.0226	0.1765	0.0070
	$T_a = 50$	0.7058	0.1295	0.1135	0.1097	0.0114	0.0094	0.0139	0.0217	0.1831	0.0051
	$T_a = 20$	0.6950	0.1250	0.0464	0.1040	0.0081	0.0098	0.0152	0.0259	0.1943	0.0041
	$T_a = 5$	0.2729	0.0665	0.0319	0.0154	0.0082	0.0123	0.0151	0.0246	0.1946	0.0046
	$T_a = 2$	0.1708	0.0148	0.0116	0.0138	0.0105	0.0130	0.0165	0.0341	0.1918	0.0075
PMC	$N = 500$	30.89	16.89	13.21	2.51	0.2466	0.0215	0.0003	0.0001	0.0003	0.2565
	$N = 2000$	28.99	16.69	11.67	1.53	0.1815	0.0073	0.0001	0.0000	0.0001	0.2062

Table 7. MSE of the estimation of the normalizing constant of the target with the initialization **I1**, using MIS, PIS, APIS, MAPIS, AMIS and PMC. In Table 2.1, we set $N = 100$ ($T = 2000$; the total number of samples is $NT = 20000$) for MIS, PIS, APIS and PMC. Recall that, with AMIS, we have one proposal each iteration. Moreover, for AMIS, we show directly the best results, obtained varying M and K such that $MK = NT = 20000$ (as detailed in the text). In Table 2.2, the MSE of MAPIS and PMC with $N = 500, 2000$, is given.

Table 6.1

Alg. \ Std		$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 5$	$\sigma = 7$	$\sigma = 10$	$\sigma = 20$	$\sigma = 70$	$\sigma_{i,j} \sim \mathcal{U}([1, 10])$
MIS		$2.15 \cdot 10^4$	524.3044	29.6219	6.1750	0.5084	0.0644	0.0043	0.0002	0.0016	0.6935
PIS ($T_a = T$)		0.5402	0.0096	0.0001	0.0001	0.0001	0.0001	0.0001	0.0004	0.0016	0.0001
APIS	$T_a = 100$	0.0100	0.0003	0.0000	0.0000	0.0000	0.0001	0.0001	0.0002	0.0017	0.0000
	$T_a = 50$	0.0068	0.0002	0.0000	0.0000	0.0000	0.0001	0.0001	0.0002	0.0017	0.0000
	$T_a = 20$	0.0055	0.0004	0.0000	0.0000	0.0000	0.0001	0.0001	0.0002	0.0016	0.0000
	$T_a = 5$	0.0023	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001	0.0002	0.0017	0.0000
	$T_a = 2$	0.0008	0.0000	0.0001	0.0000	0.0001	0.0001	0.0002	0.0003	0.0017	0.0001
AMIS	(best)	15.92	15.66	13.92	9.80	3.06	0.4207	0.0053	0.0001	0.0001	– – – – –
	(worst)	15.95	15.93	15.18	12.51	5.92	1.93	0.4155	0.0006	1.53	– – – – –
PMC	$N = 100$	35.09	17.04	15.20	6.85	0.4115	0.0586	0.0015	0.0002	0.0016	0.3457
M-PMC	$N = 100$	1.13	0.1906	0.1103	0.2959	0.0383	0.0022	0.0003	0.0002	0.0017	0.0004

Table 6.2

MAPIS	$T_a = 100$	0.0055	0.0003	0.0000	0.0000	0.0001	0.0001	0.0001	0.0002	0.0017	0.0000
	$T_a = 50$	0.0046	0.0002	0.0000	0.0000	0.0001	0.0001	0.0001	0.0002	0.0015	0.0000
	$T_a = 20$	0.0035	0.0001	0.0000	0.0000	0.0001	0.0001	0.0001	0.0017	0.0017	0.0000
	$T_a = 5$	0.0028	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001	0.0002	0.0019	0.0000
	$T_a = 2$	0.0007	0.0001	0.0000	0.0000	0.0001	0.0001	0.0001	0.0003	0.0018	0.0001
PMC	$N = 500$	30.85	16.67	14.30	2.41	0.2571	0.0218	0.0003	0.0001	0.0003	0.2491
	$N = 2000$	26.91	16.48	11.92	1.63	0.2087	0.0059	0.0001	0.0000	0.0001	0.2039

Table 8. MSE of the estimation of the normalizing constant of the target with the initialization **I2**, using MIS, PIS, APIS, MAPIS, AMIS and PMC. In Table 2.1, we set $N = 100$ ($T = 2000$; the total number of samples is $NT = 20000$) for MIS, PIS, APIS and PMC. Recall that, with AMIS, we have one proposal each iteration. Moreover, for AMIS, we show directly the best results, obtained varying M and K such that $MK = NT = 20000$ (as detailed in the text). In Table 2.2, the MSE of MAPIS and PMC with $N = 500, 2000$, is given.