Matlab code of Layered Adaptive Importance Sampling

Luca Martino⋆, Víctor Elvira⊤, David Luengo⋄

⋆ Universitat de Valencia, Valencia (Spain).

⊤ Télécom Lille, Institut Mines-Télécom, Lille (France).

⋄ Universidad Politecnica de Madrid, Madrid (Spain).

Abstract

In this document, we present a preliminary Matlab implementation of the Layered Adaptive Importance Sampling (LAIS) scheme. We mainly focus on an non-iterative implementation but a more specific iterative version is also provided. We also introduce briefly the framework and discuss the benefit of LAIS with respect to other Monte Carlo techniques. Note that the code is not optimized.

Keywords: Adaptive Importance Sampling; MCMC; Parallel MCMC; Matlab Code.

1 Introduction

Monte Carlo methods are widely used in signal processing, communications, machine learning and many other scientific disciplines [13, 41, 1, 38, 39, 24, 19]. Importance sampling (IS) [24, 39] and MCMC algorithms [1, 11, 20, 22, 32] are well-known Monte Carlo (MC) methodology that can be used to compute integrals involving a complicated multidimensional target probability density function (pdf), $\pi(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^d$. The MCMC techniques generate a Markov chain with invariant density the target pdf, using a simpler proposal density$q(\mathbf{x})$. The IS method proceeds by drawing samples from the proposal pdf, $q(\mathbf{x})$, and assigning weights to them according to the ratio between the target and the proposal, i.e., $w(\mathbf{x}) = \frac{\pi(\mathbf{x})}{q(\mathbf{x})}$.

Benefits and issues of MCMC and IS. On the one hand, denoting the state of the chain as $\mathbf{x}_t$, an MCMC algorithm using a random walk proposal pdf $q(\mathbf{x}|\mathbf{x}_t) = q(\mathbf{x} - \mathbf{x}_t)$ has an intrinsic explorative behavior that is one of the reason of MCMC success. However, it is difficult to determine the burn-in period of the MCMC convergence. Moreover, it is complicate to estimate the marginal likelihood given the MCMC outputs and ensuring the ergodicity of and adaptive MCMC is not an easy task. On the other hand, the IS approach easily provides and estimation of the marginal likelihood. Although the validity of the IS approach is guaranteed under mild assumptions, the variance of the estimator depends critically on the discrepancy between the proposal and the target densities. An adaptive strategy is required to provide robust results (the theoretical analysis for adaptive IS is usually simpler than in adaptive MCMC schemes).

Adaptive Importance Sampling. In order to solve this issue, several works are devoted to the design of adaptive IS (AIS) schemes [6], where the proposal density is updated by learning from previously generated samples [39]. Some well-known methods available in the literature are Population Monte Carlo (PMC) [8, 7, 21, 17], Adaptive Multiple Importance Sampling (AMIS) [10], Adaptive Population Importance Sampling (APIS) [26, 27], or Layered Adaptive Importance Sampling (LAIS) [30, 29, 28] and other alternatives using gradient information of the target pdf [18, 40].

All of them are iterated importance samplers, and most of them employ the multiple importance sampling (MIS) approach [15, 14, 16] for building the IS estimators. In all of these schemes, the proposal or the population of proposals is updated at each iteration, using the statistical information that has been collected from the target in the previous iterations through the samples generated. However, the aforementioned methods differ substantially, both in the specific MIS scheme used to perform the estimation (different weighting functions can be employed within a MIS framework, as shown in [15]) and in the adaptation procedure used to update the proposals.

Layered Adaptive Importance Sampling (LAIS) [30]. LAIS is formed by two Monte Carlo levels. An upper layer produces MCMC outputs, that are used as parameters in an IS scheme (lower layer). More specifically, the MCMC outputs are employed as means of the proposal pdfs used in the lower layer, i.e., in the IS scheme. In its iterative version, LAIS can be interpreted as an adaptive importance sampling driven by an MCMC procedure. However, considering a non-iterative implementation, it can be also seen as a static importance sampling using multiple different proposal pdfs, whose means has been previously selected by some MCMC procedure.

> LAIS combines the benefits of the IS and MCMC approaches: explorative behavior, automatic selection of the location parameters of the algorithm, and easy estimation of the marginal likelihood.

> A specific LAIS algorithm is specified by the specific choice of the upper layer (i.e., the MCMC procedure) and by the specific choice the lower layer (i.e., the MIS procedure).

Other attempts have been made to successfully marry the IS and MCMC approaches, producing hybrid techniques: IS-within-MCMC [2, 5, 23, 25, 35, 33] or MCMC-within-IS [3, 4, 9, 34, 36, 37, 42]. Related reviews and tutorials about IS, AIS, and Multiple IS (MIS) can be found at [6, 15, 31]. Some related slides can be found at

`http://www.lucamartino.altervista.org/LAIS_pres_Completa.pdf`

> In this Matlab implementation, we focus on independent parallel Metropolis-Hastings (MH) chains in upper level, and we consider three different MIS strategies.

The Matlab code of the non-iterative version of LAIS can be found at

`http://www.mathworks.com/matlabcentral/fileexchange/`

59952-non-iterative-implementation-of-layered-adaptive-importance-sampling--lais-

or

http://www.lucamartino.altervista.org/LAIS_non_iterative_code.zip

The Matlab code of a specific iterative implementation of LAIS can be found at

http://www.mathworks.com/matlabcentral/fileexchange/
58368-parallel-interacting-markov-adaptive-importance-sampling--pi-mais--algorithm

or

http://lucamartino.altervista.org/CODE_LAIS_v03.zip

Note that the code is not optimized.

2 Static Importance Sampling

Let us consider the variable of interest, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$, and the target density

$$\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x}), \tag{1}$$

that usually corresponds to a posterior pdf, i.e., the density of the variable of interest $\mathbf{x}$ given the collected data $\mathbf{y}$,

$$\bar{\pi}(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}).$$

Our goal is computing efficiently some moment of $\mathbf{x}$,

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

and the so-called *marginal-likelihood* (a.k.a., *Bayesian evidence*) [39],

$$Z = p(\mathbf{y}) = \int_{\mathcal{X}} \pi(\mathbf{x})\mathbf{dx}, \tag{3}$$

useful for tuning parameters or, more in general, for model selection purpose. We consider $f(\mathbf{x})$ can be any square-integrable function. In general, we are not able to draw samples from $\bar{\pi}(\mathbf{x})$. The importance sampling technique is based on the following equality

$$I = E_q[w(\mathbf{X})f(\mathbf{X})] = \frac{1}{Z}\int_{\mathcal{X}} f(\mathbf{x})w(\mathbf{x})q(\mathbf{x})d\mathbf{x}, \tag{4}$$

where q is a proposal pdf (simpler to draw from and with heavier tails than $\bar{\pi}$), and

$$w(\mathbf{x}) = \frac{\pi(\mathbf{x})}{q(\mathbf{x})}. \tag{5}$$

Hence, drawing M samples $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}$ from $q(\mathbf{x})$, the standard IS estimator when Z is known is given by

$$\widehat{I} = \frac{1}{M}\frac{1}{Z}\sum_{m=1}^{M} w^{(m)} f(\mathbf{x}^{(m)}), \tag{6}$$

where $w^{(m)} = w(\mathbf{x}^{(m)}) = \frac{\pi(\mathbf{x}^{(m)})}{q(\mathbf{x}^{(m)})}$. Alternatively, when Z is unknown the standard *self-normalized* IS estimator is

$$\widetilde{I} = \sum_{m=1}^{M} \bar{w}^{(m)} f(\mathbf{x}^{(m)}), \tag{7}$$

where $\bar{w}^{(m)} = \frac{w^{(m)}}{\sum_{\ell=1}^{M} w^{(\ell)}}$.

2.1 Optimal proposal density

Given a specific function $f(\mathbf{x})$, the optimal proposal q that minimizes the variance of the IS estimator is $q(\mathbf{x}) \propto |f(\mathbf{x})|\bar{\pi}(\mathbf{x})$ (at least, when $\widehat{I}$ is applied). However, in practical applications we are often interested in computing expectations w.r.t. several f's. Namely, we are interesting in a numerical approximation of $\bar{\pi}$. In this context, a more appropriate strategy is minimizing the variance of the importance weights $w^{(m)}$ for $m = 1, \dots, M$. In this case, the minimum variance is attained when $q(\mathbf{x}) = \bar{\pi}(\mathbf{x})$ [12].

2.2 Multiple Importance Sampling

As explained above, we are interested in choosing the proposal pdf q as close as possible to $\bar{\pi}$. However, K different proposal pdfs can be also considered, $q_1, \dots, q_K$ [15]. In this case, drawing KM samples

$$\{\mathbf{x}_k^{(m)}\}_{m=1}^{M} \sim q_k(\mathbf{x}), \qquad k = 1, \dots, K,$$

there are several valid choices of weights

$$w^{(m)} = \frac{\pi(\mathbf{x}_n^{(m)})}{\Phi_k(\mathbf{x}_k^{(m)})}. \tag{8}$$

Namely, several choices of the denominator $\Phi_k(\mathbf{x})$ that lead to an unbiased estimator $\widehat{I}$ in Eq. (6) are possible, as shown in [15]. Two examples are the *standard IS* weights, i.e.,

$$\Phi_k(\mathbf{x}_k^{(m)}) = q_k(\mathbf{x}^{(m)}) \Longrightarrow w^{(m)} = \frac{\pi(\mathbf{x}_n^{(m)})}{q_k(\mathbf{x}^{(m)})}, \tag{9}$$

and the *Full Deterministic Mixture (DM)* approach, i.e.,

$$\Phi_k(\mathbf{x}_k^{(m)}) = \frac{1}{K}\sum_{j=1}^{K} q_j(\mathbf{x}_k^{(m)}) \Longrightarrow w^{(m)} = \frac{\pi(\mathbf{x}_n^{(m)})}{\frac{1}{K}\sum_{j=1}^{K} q_j(\mathbf{x}_k^{(m)})}. \tag{10}$$

The full DM strategy provides better performance but it is more costly than the standard IS approach [15]. Other intermediate solutions called *partial DM* strategies [14, 16] are possible, which in general provide good compromises between performance and computational cost.

3 Matlab implementation of Layered Adaptive Importance Sampling (LAIS)

A generic adaptive importance sampler is an iterative algorithm where a set of N proposals are adapted over time, $t = 1, \dots, T$. When a *population* of proposal pdfs is employed, at the t-th iteration, the set of available proposals is $\{q_{1,t}, \dots, q_{N,t}\}$. These proposals can change over time, not only updating their parameters but even completely changing their shape (i.e., the family of the distribution), and we assume that all the $q_{n,t}$'s have heavier tails than the target $\bar{\pi}$ [39]. We only consider location-scale densities such that each $q_{n,t}$ is completely characterized by a mean vector, $\boldsymbol{\mu}_{n,t}$, and a covariance matrix, $\mathbf{C}_n$. In LAIS, we focus on the adaptation of the mean vectors, i.e., the $\boldsymbol{\mu}_{n,t}$'s. Table 1 summarizes a non-iterative version of LAIS.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%   NON-ITERATIVE IMPLEMENTATION OF               %%%%%%%
%%%%%%%   LAYERED ADAPTIVE IMPORTANCE SAMPLING (LAIS)   %%%%%%%
%%%%%%%      (see reference below)                      %%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% L. Martino, V. Elvira, D. Luengo, J. Corander,
%%%"Layered Adaptive Importance Sampling", Statistics and Computing, 2016.
%%% doi:10.1007/s11222-016-9642-5
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
clc
help MAIN
%%%
typeTar=1;%%% typeTar: type of the  target distribution
          %%% typeTar=1,2,3 =>DIM=2
          %%% typeTar=4 =>DIM=4
          %%% typeTar=5 =>DIM=10
          %%% Build your inference problem, changing target.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% upper layer: parallel MH chains  %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% number of parallel chains
N=100; %%% N>=1
%%%%  number of vertical and horizontal steps per epoch
T=20; %%% T>=1
sig_prop=5; %%std of the proposal pdfs of upper layer
[mu_tot,mu_sp,mu_time]=Upper_Layer_ParMH(N,T,sig_prop,typeTar);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% lower layer: MIS schemes         %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M=100; %%% M>=1 %%% samples per proposal pdfs in the lower layer
%sig_lower_layer=3+7*rand(1,N*T);%% possible random selection of the scalar ...
   parameters
```

Table 1: **Non-iterative version of LAIS**

Initialization: Choose

1. N = number of different proposal pdfs (in the lower layer) per iteration.
2. T = iterations of the MCMC algorithms in the upper layer.
3. $q_{n,0}$ proposal pdfs with means $\boldsymbol{\mu}_{n,0}$ and covariance matrices $\mathbf{C}_n$, $n = 1, \dots, N$.

Upper Layer - Adaptation of the means: Apply T-steps of a suitable MCMC procedure to update the mean vectors,

$$\{\boldsymbol{\mu}_{n,0}\}_{n=1}^N \longrightarrow \{\boldsymbol{\mu}_{n,1}\}_{n=1}^N \longrightarrow \cdots \{\boldsymbol{\mu}_{n,t}\}_{n=1}^N \longrightarrow \{\boldsymbol{\mu}_{n,t+1}\}_{n=1}^N \longrightarrow \cdots \{\boldsymbol{\mu}_{n,T}\}_{n=1}^N.$$

Lower Layer - MIS scheme:

1. **Sampling:** Draw M samples $\mathbf{x}_{n,t}^{(1)}, \dots, \mathbf{x}_{n,t}^{(M)}$ from each of the N proposal pdfs in the population $\{q_1, \dots, q_N\}$, i.e.,
$$\mathbf{x}_{n,t}^{(m)} \sim q_{n,t}(\mathbf{x}|\boldsymbol{\mu}_{n,t}, \mathbf{C}_n),$$
for $m = 1, \dots, M$.
2. **Weighting:** Weight the samples, $\{\mathbf{x}_{n,t}^{(m)}\}_{m=1}^M$,
$$w_{n,t}^{(m)} = \frac{\pi(\mathbf{x}_{n,t}^{(m)})}{\Phi_{n,t}(\mathbf{x}_{n,t}^{(m)})}, \qquad (11)$$
according to a proper function $\Phi_{n,t}$ that leads to unbiased IS estimators $\hat{I}$ [24, Section 2.5.4][30, 15].

Output: Return $\{\mathbf{x}_{n,t}^{(m)}, w_{n,t}^{(m)}\}$ for all $n = 1, \dots, N$, $m = 1, \dots, M$ and $t = 1, \dots, T$. The normalized weights will be

$$\bar{w}_{n,t}^{(m)} = \frac{w_{n,t}^{(m)}}{\sum_{\tau=1}^T \sum_{j=1}^N \sum_{r=1}^M w_{j,\tau}^{(r)}}. \qquad (12)$$

```
sig_lower_layer=13*ones(1,N*T);
%%%%%%%%%%
%%% SUGGESTION: USE typeDEN=3 as in PI-MAIS (see article)
typeDEN=3; %%%% type of the MIS scheme
%%%% 1 - Standard IS
%%%% 2 - Full DM
%%%% 3 - Partial DM - spatial (suggested)
if typeDEN==1
  disp(' ')
  disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
  disp('We suggest to use the Partial DM approach (typeDEN=3)')
```

```
  disp('and to avoid typeDEN=1, specially for the estimation of the marginal ...
     likelihood')
  disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
  disp(' ')
  pause(1)
end
[x_est,MarginalLike,x_IS,W]=Lower_Layer_IS(mu_tot,mu_sp,mu_time,N,T,M, ...
sig_lower_layer,typeDEN,typeTar);
```

Listing 1: MAIN.m (non-iterative implementation).

Upper Layer: Some MCMC procedure is applied to obtain all the means $\{\boldsymbol{\mu}_{n,t}\}$ with $n = 1, \ldots, N$, and $t = 1, \ldots, T$. In this implementation, we have considered *independent parallel Metropolis-Hastings (MH) chains*. In this scenario, N is also the number of parallel chains.

In the upper layer, the employed MCMC scheme can address a target density different from the posterior pdf (for instance, a tempered target). In this specific implementation, we have considered MCMC addressing the true target pdf $\bar{\pi}(\mathbf{x})$ (but other choices are possible).

In LAIS, the mean parameters selected the upper layer are chosen independently from the Monte Carlo outputs in the lower layer.

The total number of mean-location parameters (and proposal pdfs) is NT.

```
function [x_tot,x,x_t]=Upper_Layer_ParMH(N,T,sigprop,typeTar)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% LAIS - upper level               %%%%%%%
% %% parallel MH chains               %%%%%%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % L. Martino, V. Elvira, D. Luengo, J. Corander,
% %"Layered Adaptive Importance Sampling",
% % Statistics and Computing, 2016. doi:10.1007/s11222-016-9642-5
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % N= number of chain
% % sigprop = std of the proposal pdfs of the vertical chains
% % x_tot= generated samples (states of the chains)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if N<2
    N=2;
end
if T<1
    T=1;
end
   disp('------------------------------------------------------------')
   disp('****-****-****-****-****-****-****-****')
```

```
   disp('****       UPPER-LAYER            *****')
   disp('****-****-****-****-****-****-****-****')
   disp('-----------------------------------------------------------')
    disp(['Number of chains= ' num2str(N), ' '])
    disp(['Iterations= ' num2str(T)])
    disp(['Total number of generated mean parameters= ' num2str(N*T)])

   disp('-----------------------------------------------------------')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% TARGET %%%%%%%%%%%%%%%%%%%%%
 %%%%typeTar: type of the  target distribution %%% change target.m
[nothing,nothing,DIM]=target(NaN,typeTar); %%%%%% DIM= dimension
logf=@(x) target(x,typeTar);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% parameters %%%%%%%%%%%%%%%%%%%%%
initialPoints=-4+8*rand(N,DIM);
x{1}=initialPoints;
SIGMA=sigprop.^2*eye(DIM);
x_tot=[];
 Vbef=logf(x{1});
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%% START %%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%% PARALLEL  CHAINS                   %%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  for t=2:T+1
     x{t}=mvnrnd(x{t-1},SIGMA);
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  %%% evaluating target %%%%%%%%%%%%
   %%%% Vbef=logf(x{t-1}); %%% THIS EVALUATION CAN BE AVOIDED, storing the ...
      previous one
    Vnow=logf(x{t});
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [d1,d2]=size(Vbef);
     if d1>d2
         Vbef=Vbef';
         Vnow=Vnow';
     end
    rho=exp(Vnow-Vbef);
    alpha=min([ones(1,N);rho]);
    %%% Test %%%
    u=rand(1,N);
    test=(u<=alpha);
    vec_aux=(x{t}-x{t-1}).*repmat(test',1,DIM);
    vec_aux2=(Vnow-Vbef).*test;
    x{t}=x{t-1}+vec_aux;
```

```
    Vbef=Vbef+vec_aux2;
    Vbef=Vbef';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      x_tot=[x_tot x{t}'];
 end %%% end for - t
x{1}=[];
x=x(~cellfun('isempty',x));
   for i=1:N
    x_t{i}=x_tot(:,i:N:end);
   end
```

Listing 2: Upper-Layer-ParMH.m (non-iterative implementation).

Lower Layer: M independent samples are drawn from each proposal in Step 1 of Table 1, i.e.,

$$\mathbf{x}_{n,t}^{(m)} \sim q_{n,t}(\mathbf{x}|\boldsymbol{\mu}_{n,t}, \mathbf{C}_n), \tag{13}$$

for $m = 1, \ldots, M$, $n = 1, \ldots, N$ and $t = 1, \ldots, T$. Then the weight

$$w_{n,t}^{(m)} = \frac{\pi(\mathbf{x}_{n,t}^{(m)})}{\Phi_{n,t}(\mathbf{x}_{n,t}^{(m)})}, \tag{14}$$

is associated to each one of these NMT samples. In the denominator, we have a function $\Phi_{n,t}$, properly chosen in such a way that the estimator of Eq. (6) is unbiased [24, Section 2.5.4], [30, 15].

> The total number of weighted samples involved in the IS estimators are $K = NMT$.

In this non-iterative MATLAB code of LAIS,
`http://www.lucamartino.altervista.org/LAIS_non_iterative_code.zip`
we have implemented 3 possible denominator functions $\Phi_{n,t}(\mathbf{x}_{n,t}^{(m)})$:

- `typeDEN=1`: standard IS approach,

$$\Phi_{n,t}(\mathbf{x}_{n,t}^{(m)}) = q_{n,t}(\mathbf{x}_{n,t}^{(m)}|\boldsymbol{\mu}_{n,t}, \mathbf{C}_n), \tag{15}$$

- `typeDEN=2`: Full DM approach,

$$\Phi_{n,t}(\mathbf{x}_{n,t}^{(m)}) = \frac{1}{NT} \sum_{\tau=1}^{T} \sum_{j=1}^{N} q_{j,\tau}(\mathbf{x}_{n,t}^{(m)}|\boldsymbol{\mu}_{j,\tau}, \mathbf{C}_j), \tag{16}$$

- `typeDEN=3`: Partial DM-Spatial approach,

$$\Phi_{n,t}(\mathbf{x}_{n,t}^{(m)}) = \frac{1}{N} \sum_{j=1}^{N} q_{j,t}(\mathbf{x}_{n,t}^{(m)}|\boldsymbol{\mu}_{j,t}, \mathbf{C}_j). \tag{17}$$

The choice **typeDEN=1** is the faster alternative but it also provides the worst performance (specially, in multimodal scenarios).

The choice **typeDEN=2** provides the best performance but also the greatest computational cost.

The choice **typeDEN=3** is an efficient compromise between good performance and computational cost [15, 14, 16].

The specific LAIS scheme with parallel MH chains in the upper layer and a partial-spatial DM **typeDEN=3** is specially suitable for multimodal scenarios.

```
function [x_est,MarginalLike,x_IS,W]=Lower_Layer_IS(mu,mu_sp,mu_time,...
                                    N,T,M,sig_lower_layer,typeDEN,typeTar)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%% Lower Layer of LAIS %%%%%%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % L. Martino, V. Elvira, D. Luengo, J. Corander,
% %"Layered Adaptive Importance Sampling",
% % Statistics and Computing, 2016. doi:10.1007/s11222-016-9642-5
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%% typeDEN  1: 'StandIS' -- 2: 'FullDM' --- 3: 'PartialDM-Spatial'
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
Np=N*T;
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
   disp('-----------------------------------------------------------')
   disp('****-****-****-****-****-****-****-****')
   disp('****       LOWER-LAYER            *****')
   disp('****-****-****-****-****-****-****-****')
   disp('-----------------------------------------------------------')
    disp(['Number of proposal pdfs used in MIS = ' num2str(Np)  ' '])
    disp(['Number of samples per proposal= ' num2str(M)])
    disp(['Total number of used samples = ' num2str(Np*M)])
    switch typeDEN
        case 1
            aux='Standard IS';
            NumDen=1;
            NumProp=1;
        case 2
             aux='Full DM';
             NumDen=1;
             NumProp=N*T;
         case 3
             aux='Partial DM - Spatial';
```

```
            NumDen=T;
            NumProp=N;
    end
    disp(' ')
    disp(['Type IS-Denominator= ' aux])
    disp(['Number of different denominators = ' num2str(NumDen)])
    disp(['Number of proposal pdfs in each denominator = ' num2str(NumProp)])

    disp('---------------------------------------------------------')
%%%% only for getting the dimension
[nothing,nothing,DIM,mu_true,Marglike_true]=target(NaN,typeTar); %%%%%% DIM= ...
   dimension
%%%%  target distribution %%% change target.m
logf=@(x) target(x,typeTar);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% normalizing constant proposal ==> 1/(sqrt((2*pi).^DIM*det(SIGMA_p)))
count=1;
t=1;
n=1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:Np
%%%% generated samples - lower layer %%%%%%%%%%%%
SIGMA_p=sig_lower_layer(j)*eye(DIM);
x_IS(:,M*(j-1)+1:M*j)=mvnrnd(mu(:,j),SIGMA_p,M)';

%%%%%% %%%%%% %%%%%%
if typeDEN==1 %%% 'StandIS'
%%%%% %%%% %%%% %%%% %%%% %%%% %%%% %%%% %%%%  %%%%
%%%% evaluate j-th proposal - only own samples %%%%
%%%% %%%% %%%% %%%% %%%% %%%% %%%% %%%% %%%% %%  %%
P(M*(j-1)+1:M*j)=mvnpdf(x_IS(:,M*(j-1)+1:M*j)',mu(:,j)',SIGMA_p);
end
%%%%%% %%%%%% %%%%%%
if typeDEN==3
   x_ISsp{t}(:,M*(count-1)+1:M*count)=x_IS(:,M*(j-1)+1:M*j);
   count=count+1;
    if mod(j,N)==0
     t=t+1;
     count=1;
   end
end
%%%%%% %%%%%% %%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% evaluate target          %%%%
logNUM(M*(j-1)+1:M*j)=logf(x_IS(:,M*(j-1)+1:M*j)')';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
switch  typeDEN
    %%%%%%%%%%%%%%%%%%%%%%
    case 1 %%% 'StandIS'
      logDEN=log(P);
    %%%%%%%%%%%%%%%%%%%%%%
    case 2 %%% 'FullDM' (evaluating 'All-in-All')
    %%%%%%%%%%%%%%%%%%%%%%
        for j=1:Np

             if mod(j,10)==0
                      disp(['computing IS ...
                         denominators...',num2str(min([fix(j*100/Np) 100])) '%'])
             end

            for k=1:Np
               P(k,M*(j-1)+1:M*j)=mvnpdf(x_IS(:,M*(j-1)+1:M*j)',mu(:,k)',SIGMA_p);
               %%% P size N*T \times N*T*M
               %%% row => evaluation one proposal pdf at all the samples
               %%% (row index=> denote one proposal pdf)
               %%% (column index=> denote one sample)
            end
        end
          P=mean(P);
          logDEN=log(P);
      %%%%%%%%%%%%%%%%%%%%%%
       case 3 %%% 'Partial DM- Spatial'
      %%%%%%%%%%%%%%%%%%%%%%
          for t=1:T
                 if mod(t,10)==0
                      disp(['computing IS ...
                         denominators...',num2str(min([fix(t*100/T) 100])) '%'])
                 end

              for n=1:N
                  Maux(n,:)=mvnpdf(x_ISsp{t}',mu_sp{t}(n,:),SIGMA_p);
              end
              a(t,:)=log(mean(Maux));
          end
           logDEN= reshape(a',1,N*T*M);
end %%% build denominator
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% WEIGHTING %%%%%%%%%%%
W=exp(logNUM-logDEN);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% ESTIMATIONS %%%%%%%%%
MarginalLike=mean(W); %%% estimation of the marginal likelihood
disp(' ')
disp('RESULTS')
disp(' ')
disp(['Marginal likelihood- true value = ', num2str(Marglike_true)])
disp(['Marginal likelihood- estimated value = ', num2str(MarginalLike)])
```

```
SEmargLike=(Marglike_true-MarginalLike)^2;
disp(['Square Error in the estimation of marginal likelihood = ', ...
   num2str(SEmargLike)])
%%%%%%%%%%%%
Wn=W./(Np*M*MarginalLike);
x_est=sum(repmat(Wn,DIM,1).*x_IS,2);
disp(' ')
disp(['Expected Value of the posterior/target pdf - True Values = ...
   ',num2str(mu_true')])
disp(['Expected Value of the  posterior/target pdf-Estimated Values= ...
   ',num2str(x_est')])
SE_est=mean((mu_true-x_est).^2);
disp(['Square Error in the estimation of the Expected Value = ', num2str(SE_est)])
%%%%%%%%%%%%%%
%%%% plot %%%%
if typeTar==1 | typeTar==2 | typeTar==3
 if typeTar==1
    hgload('contourGauss5modes.fig');
 end
 hold on
  plot(x_IS(1,:),x_IS(2,:),'g.','MarkerEdgeColor','g','MarkerFaceColor','g',...
      'MarkerSize',1)
 plot(mu(1,:),mu(2,:),'rs','MarkerEdgeColor','k','MarkerFaceColor','r',...
     'MarkerSize',5)
  axis([-22 22 -25 25])
end
```

Listing 3: Lower-Layer-IS.m (non-iterative implementation).

A specific iterative implementation of LAIS (employing parallel MH chains and `typeDEN=3`) can be found at
`http://lucamartino.altervista.org/CODE_LAIS_v03.zip`

4 Numerical comparison

Let us consider a bivariate multimodal target pdf formed by a mixture of 5 Gaussians, i.e.,

$$\bar{\pi}(\mathbf{x}) = \frac{1}{5}\sum_{i=1}^{5}\mathcal{N}(\mathbf{x};\boldsymbol{\nu}_i,\boldsymbol{\Sigma}_i), \quad \mathbf{x}\in\mathbb{R}^2, \tag{18}$$

with means $\boldsymbol{\nu}_1 = [-10,-10]^\top$, $\boldsymbol{\nu}_2 = [0,16]^\top$, $\boldsymbol{\nu}_3 = [13,8]^\top$, $\boldsymbol{\nu}_4 = [-9,7]^\top$, and $\boldsymbol{\nu}_5 = [14,-14]^\top$, and covariance matrices $\boldsymbol{\Sigma}_1 = [2,\ 0.6; 0.6,\ 1]$, $\boldsymbol{\Sigma}_2 = [2,\ -0.4; -0.4,\ 2]$, $\boldsymbol{\Sigma}_3 = [2,\ 0.8; 0.8,\ 2]$, $\boldsymbol{\Sigma}_4 = [3,\ 0; 0,\ 0.5]$, and $\boldsymbol{\Sigma}_5 = [2,\ -0.1; -0.1,\ 2]$.

> This target corresponds to the choice `typeTar=1` in the code of the non-iterative LAIS version.

The goal is to estimate, via Monte Carlo, the mean of $\bar{\pi}$ (the true value is $[1.6, 1.4]^\top$), using different adaptive algorithms. For the sake of simplicity, we also consider Gaussian proposals.

Table 2: MSE in the estimation of the mean of the target. The best results for each value of σ are highlighted in bold-face.

Algorithm	$\boldsymbol{\sigma = 1}$	$\boldsymbol{\sigma = 5}$	$\boldsymbol{\sigma = 10}$	$\boldsymbol{\sigma_{i,j} \sim \mathcal{U}([1, 10])}$
PMC [8]	107.58	0.6731	0.0744	0.0732
AMIS [10]	121.21	0.8640	**0.0121**	0.7328
APIS [27]	2.45	0.2424	0.0185	0.0045
LAIS [30]	**0.0021**	**0.0075**	**0.0121**	0.0049
GAPIS [18]	0.0023	0.0104	0.0143	**0.0040**

We have tested APIS, AMIS, standard PMC, gradient APIS (GAPIS), and LAIS. At each run, the initial means of the proposals are selected uniformly within a square $\boldsymbol{\mu}_i^{(0)} \sim \mathcal{U}([-4, 4] \times [-4, 4])$. Initially, the same isotropic covariance matrix, $\mathbf{C}_i^{(0)} = \sigma^2 \mathbf{I}_2$, is used for every proposal. We test different values of $\sigma \in \{1, 5, 10\}$. Then, different non-isotropic diagonal covariance matrices, $\mathbf{C}_i = \text{diag}(\sigma_{i,1}^2, \sigma_{i,2}^2)$ with $\sigma_{i,j} \sim \mathcal{U}([1, 10])$ for $i = 1, \ldots N$ and $j \in \{1, 2\}$, are also tested. In the upper layer in LAIS, we use random walk Gaussian proposal pdfs with the covariance matrices as in the lower layer, i.e., $\mathbf{C}_i$.

In this example, all the algorithms adapt only the means of the proposals, except for AMIS, that also adapts the covariance matrix. We use $N = 100$ for all the methods, except for AMIS (we recall that $N = 1$ in this case). In order to obtain a fair comparison, for each method T is selected in such a way that the total number of evaluations of the target is always fixed to $E = 2 \cdot 10^5$. Table 2 shows the mean squared error (MSE) of the estimation of the first component of the mean (averaged over 500 runs). We have considered the best performance of each method after testing several combinations of parameters, as described in [27, 30, 18]. Note that GAPIS and LAIS provide the best results in general, whereas AMIS and APIS are also competitive when $\sigma = 10$ and $\sigma_{i,j} \sim \mathcal{U}([1, 10])$, respectively.

The advantages of applying LAIS are even more evident in the estimation of the marginal likelihood (which is also a very hard task using MCMC).

References

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

[2] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society B*, 72(3):269–342, 2010.

[3] F. Beaujean and Caldwell A. Initializing adaptive importance sampling with Markov chains. *arXiv:1304.7808*, 2013.

[4] Z. I. Botev, P. LEcuyer, and B. Tuffin. Markov chain importance sampling with applications to rare event probability estimation. *Statistics and Computing*, 23:271–285, 2013.

[5] A. Brockwell, P. Del Moral, and A. Doucet. Sequentially Interacting Markov chain Monte Carlo methods. *The Annals of Statistics*, 38(6):3387–3411, 2010.

[6] M. F. Bugallo, L. Martino, and J. Corander. Adaptive importance sampling in signal processing. *Digital Signal Processing*, 47:36–49, 2015.

[7] 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.

[8] 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.

[9] N. Chopin and C. P. Robert. Properties of nested sampling. *Biometrika*, 97(3):741–755, 2010.

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

[11] J. Dagpunar. *Simulation and Monte Carlo: With applications in finance and MCMC*. John Wiley and Sons, Ltd, England, 2007.

[12] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12(3):656–704, 2009.

[13] A. Doucet and X. Wang. Monte Carlo methods for signal processing. *IEEE Signal Processing Magazine*, 22(6):152–170, Nov. 2005.

[14] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Efficient multiple importance sampling estimators. *Signal Processing Letters, IEEE*, 22(10):1757–1761, 2015.

[15] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. Generalized multiple importance sampling. *arXiv:1511.03095*, 2015.

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

[17] V. Elvira, L. Martino, D. Luengo, and M.F. Bugallo. Improving Population Monte Carlo: Alternative weighting and resampling schemes. *Signal Processing*, 131(656-704):77–91, 2017.

[18] V. Elvira, L. Martino, D. Luengo, and J. Corander. A gradient adaptive population importance sampler. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4075–4079, 2015.

[19] W. Fong, S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing with application to audio signal enhancement. *IEEE Transactions on Signal Processing*, 50(2):438–448, February 2002.

[20] Y. Fong, J. Wakefield, and K. Rice. An efficient Markov Chain Monte Carlo method for mixture models by neighborhood pruning. *Journal of Computational and Graphical Statistics (to appear)*, 2011.

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

[22] J. R. Larocque and P. Reilly. Reversible jump mcmc for joint detection and estimation of sources in colored noise. *IEEE Transactions on Signal Processing*, 50(2), February 1998.

[23] R. Liesenfeld and J. F.Richard. Improving MCMC, using efficient importance sampling. *Computational Statistics and Data Analysis*, 53:272–288, 2008.

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

[25] J. S. Liu, F. Liang, and W. H. Wong. The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, March 2000.

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

[27] 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.

[28] L. Martino, V. Elvira, D. Luengo, and J. Corander. Interacting Parallel Markov Adaptive Importance Sampling. *European Signal Processing Conference (EUSIPCO)*, 2015.

[29] L. Martino, V. Elvira, D. Luengo, and J. Corander. MCMC-driven adaptive multiple importance sampling. *Interdisciplinary Bayesian Statistics Springer Proceedings in Mathematics & Statistics (Chapter 8)*, 118:97–109, 2015.

[30] L. Martino, V. Elvira, D. Luengo, and J. Corander. Layered adaptive importance sampling. *Statistics and Computing, doi:10.1007/s11222-016-9642-5*, 2016.

[31] L. Martino, V. Elvira, D. Luengo, and F. Louzada. Adaptive Population Importance Samplers: A general perspective. *IEEE Sensor Array and Multichannel Signal Processing Workshop, (SAM)*, 2016.

[32] L. Martino, V. P. Del Olmo, and J. Read. A multi-point Metropolis scheme with generic weight functions. *Statistics & Probability Letters*, 82(7):1445–1453, 2012.

[33] L. Martino and J. Read. On the flexibility of the design of Multiple Try Metropolis schemes. *Computational Statistics*, 28(6):2797–2823, 2013.

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

[35] R. Neal. MCMC using ensembles of states for problems with fast and slow variables such as Gaussian process regression. *arXiv:1101.0387*, 2011.

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

[37] A. L. Nguyen. Regenerative Markov chain importance sampling. *Communications in Statistics - Simulation and Computation*, DOI: 10.1080/03610918.2015.1043383, 2015.

[38] J. Read, L. Martino, and D. Luengo. Efficient Monte Carlo optimization for multi-label classifier chains. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

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

[40] I. Schuster. Gradient importance sampling. *arXiv:1507.05781*, pages 313–316, 2015.

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

[42] X. Yuan, Z. Lu, and C. Z. Yue. A novel adaptive importance sampling algorithm based on Markov chain and low-discrepancy sequence. *Aerospace Science and Technology*, 29:253–261, 2013.