Fasiolo, M., Maskell, S., & Eler de Melo, F. (2018). Langevin incremental mixture importance sampling. *Statistics and Computing*, *28*(3), 549-561. https://doi.org/10.1007/s11222-017-9747-5

Noname manuscript No.
(will be inserted by the editor)

Langevin Incremental Mixture Importance Sampling

Matteo Fasiolo · Flávio Eler de Melo · Simon Maskell

Received: date / Accepted: date

Abstract This work proposes a novel method through which local information about the target density can be used to construct an efficient importance sampler. The backbone of the proposed method is the Incremental Mixture Importance Sampling (IMIS) algorithm of Raftery and Bao (2010), which builds a mixture importance distribution incrementally, by positioning new mixture components where the importance density lacks mass, relative to the target. The key innovation proposed here is to construct the mean vectors and covariance matrices of the mixture components by numerically solving certain differential equations, whose solution depends on the local shape of the target log-density. The new sampler has a number of advantages: a) it provides an extremely parsimonious parametrization of the mixture importance density, whose configuration effectively depends only on the shape of the target and on a single free parameter representing pseudo-time; b) it scales well with the dimensionality of the target; c) it can deal with targets that are not log-concave. The performance of the proposed approach is demonstrated on two synthetic non-Gaussian densities, one being defined on up to eighty dimensions, and on a Bayesian logistic regression model, using the Sonar dataset. The `Julia` code implementing the importance sampler proposed here can be found at https:/github.com/mfasiolo/LIMIS.

Keywords Importance sampling · Langevin diffusion · Mixture density · Optimal importance distribution · Local approximation · Kalman-Bucy filter

This work was partially funded by the Defence Science and Technology Laboratory through projects WSTC0058 and CDE36610.

Matteo Fasiolo
University of Bristol
School of Mathematics
Bristol, United Kingdom
E-mail: matteo.fasiolo@bristol.ac.uk

Flávio Eler de Melo and Simon Maskell
University of Liverpool
School of Electrical Engineering, Electronics and Computer Science
Liverpool, United Kingdom

1 Introduction

The efficiency gains brought about by taking into account local information about the target density have been amply demonstrated in the context of Markov chain Monte Carlo (MCMC) sampling. For instance, the seminal paper of Girolami and Calderhead (2011) introduced variations of the Metropolis adjusted Langevin (MALA) (Roberts and Tweedie, 1996) and Hamiltonian Monte Carlo (HMC) (Duane et al, 1987) samplers which, by exploiting second order information, can efficiently sample highly dimensional non-Gaussian targets. This is achieved using an adaptive proposal, based on the local information contained in the gradient and Hessian of the target log-density. Notably, the state-of-the-art probabilistic programming language `Stan` (Carpenter et al, 2017), uses the tuned HMC algorithm proposed by Hoffman and Gelman (2014) as its default sampler. This demonstrates that these ideas have changed MCMC sampling practice, as well as theory. It is therefore surprising that these concepts have not been exploited nearly as widely in the context of Importance Sampling (IS).

In this paper we attempt to fill this gap, by extracting local information about the target density and using it to set up an efficient importance sampler. We accomplish this by considering ideas related to Langevin

diffusions and adapting them to the context of IS. In particular, we demonstrate how linearized solutions to Langevin diffusions can produce Gaussian densities that often represent accurate local approximations to the target density. These local densities can then be combined to form a global mixture importance density that closely approximates the target. To achieve this, we exploit the Incremental Mixture Importance Sampling (IMIS) algorithm, originally proposed by Raftery and Bao (2010). This is an automatic and non-parametric approach to IS, which constructs a mixture importance density by iteratively adding mixture components in areas where the importance density lacks mass relative to the target. As the examples will demonstrate, the proposed modification of the IMIS algorithm leads to a scalable and semi-automated approach to Importance Sampling (IS).

The literature related to the current proposal is quite sparse. Indeed, the use of local target information has been adopted mostly in the context of Sequential Monte Carlo (SMC) samplers and particle filtering, rather than IS itself[1]. In particular, Sim et al (2012) and Schuster (2015) consider using MALA's adaptive proposal within SMC samplers. These proposals are quite different from our approach, because we iteratively construct a single mixture importance density, not a sequence of them. In addition, in our proposal the mean and covariance of the mixture components are not based on the derivatives of the target log-density at a single fixed location, as in MALA, but are obtained by numerically integrating certain differential equations, whose solution depends on the shape of whole regions of the target. Also, while in SMC each sample is generally perturbed individually, in our case the number of mixture components is much lower than the number of samples, which reduces the cost of constructing the importance distribution and of evaluating its density.

In the context of particle filtering, Bunch and Godsill (2016) propose a Gaussian particle flow method, which aims at approximating the optimal importance density of a class of non-linear Gaussian state space models. In particle flow algorithms (Daum and Huang, 2008) a particle is moved continuously in pseudo-time according to differential equations that depend on the underlying shape of the target density. The drawback of many particle flow algorithms is that, despite their theoretical elegance, implementing them for general models requires several layers of approximation, whose effect is not easy to quantify (Bunch and Godsill, 2016). Even though we are not considering particle filtering here, our current work has been inspired by this literature. A critical distinguishing characteristic of our proposal is that we exploit local information about the target density, while not introducing any extra approximation or source of bias in the importance sampler.

The rest of the paper is structured as follows. In Section 2 we briefly describe the IMIS algorithm of Raftery and Bao (2010). Then, in Section 3, we show how the solutions to linearized Langevin diffusions can be used to generate local approximations to the target density and we explain how these can be exploited within the IMIS algorithm. This results in the new Langevin IMIS (LIMIS) sampler. Calculating the mean vector and covariance matrix of each importance mixture component requires solving certain differential equations. This has to be done numerically, and in Section 4 we propose a novel statistically-motivated criterion for selecting the step-size of the numerical integrator. In Section 5 we compare the new sampler to IMIS, MALA and IS on three examples. Section 6 contains some discussion of the computational cost of each method, while Section 7 explains how the pseudo-time of integration can be selected in an automatic fashion. We summarize the results and discuss possible future directions in Section 8.

2 Incremental Mixture Importance Sampling

The IMIS algorithm is an automatic and non-parametric approach to IS, which is particularly useful for highly non-Gaussian target densities (Raftery and Bao, 2010). Let $\pi(\mathbf{x})$ and $p(\mathbf{x})$ be, respectively, the (possibly unnormalized) target and the prior densities, with $\mathbf{x} \in \mathbb{R}^d$. Here we describe a slightly modified version of IMIS, which includes the following steps:

Algorithm 1: Nearest Neighbour IMIS (NIMIS)

1. Initialization:
   (a) Sample n_0 variables, $\mathbf{x}_1, \ldots, \mathbf{x}_{n_0}$, from $p(\mathbf{x})$.
   (b) Calculate the weight of each sample
   $$w_i^0 = \frac{\pi(\mathbf{x}_i)}{p(\mathbf{x}_i)}, \quad \text{for} \quad i = 1, \ldots, n_0.$$
2. Importance Sampling: for $k = 1, 2, \ldots,$ repeat
   (a) Let $\mathbf{x}_j$ be the sample with the largest weight and define $\boldsymbol{\mu}_k = \mathbf{x}_j$. Calculate the covariance, $\boldsymbol{\Sigma}_k$, of the b samples with the shortest Mahalanobis distance from $\boldsymbol{\mu}_k$. The metric used to calculate the distance is the covariance of all the samples generated so far.
   (b) Generate b new samples from a multivariate Student's t distribution with mean $\boldsymbol{\mu}_k$, covariance $\boldsymbol{\Sigma}_k$ and $\nu > 0$ degrees of freedom.

[1] Note that, of course, most of the conventional SMC samplers and particle filters are based on IS.

(c) Update the importance weights of all samples generated so far

$$w_i^k = \pi(\mathbf{x}_i) \Big/ \Big\{ \frac{n_0}{n_k} p(\mathbf{x}_i) + \frac{b}{n_k} \sum_{l=1}^{k} \mathrm{mvt}(\mathbf{x}_i | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l, \nu) \Big\}, \tag{1}$$

for $i = 1, \ldots, n_k$ and where $n_k = n_0 + kb$ and $\mathrm{mvt}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$ indicates the density of a multivariate Student's t distribution, with location $\boldsymbol{\mu}$, covariance $\boldsymbol{\Sigma}$ and ν degrees of freedom.

(d) If a chosen criterion is met, terminate.

The above algorithm differs from the original IMIS procedure of Raftery and Bao (2010) in minor respects. In particular, in their version $\boldsymbol{\Sigma}_k$ is a weighted covariance, where the i-th weight is proportional to $(w_i^k + 1/n_k)/2$. We have verified that these weights can be quite unstable, especially in early iterations and in high dimensions, hence we prefer using an unweighted covariance. In step 2(a) they use the covariance of the prior distribution, rather than the covariance of all the generated samples, to determine the distances. The two approaches typically lead to similar results, but using the prior covariance can be problematic if target and prior have very different scales or correlation structures. Also, they also use multivariate Gaussian, rather than Student's t, densities. Our experience suggests that in IS it is better erring on the side of robustness, hence we prefer using Student's t densities to ensure that the proposal is heavier-tailed than the target.

The key idea behind IMIS is that it lets the importance weights determine where new mixture components should be placed. The fact that the covariance of the new components is estimated using a Nearest Neighbour approach is somewhat secondary. For this reason we use the acronym IMIS to refer to the overall approach, while we use NIMIS to refer to its Nearest Neighbour version.

In this work we use ideas related to Langevin diffusions to determine $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ in step 2(a). As we will illustrate empirically in Section 5, this modification is particularly advantageous in high dimensions. However, the purpose of this work is not so much improving upon the NIMIS algorithm, but rather showing how local information about the target can be exploited to set up an efficient mixture importance density.

3 Langevin Incremental Mixture Importance Sampling

Consider a d-dimensional Langevin diffusion, with stationary distribution $\pi(\mathbf{x})$, which is defined by the stochastic differential equation

$$d\mathbf{x}_t = \frac{dt}{2} \nabla \log \pi(\mathbf{x}_t) + d\mathbf{b}_t, \tag{2}$$

where $\nabla \log \pi(\mathbf{x})$ is the gradient of the target log-density and $\mathbf{b}_t$ is a d-dimensional Brownian motion. The dynamics of the first two moments of $\mathbf{x}_t$ are not available for most target distributions, but if we consider the discrete-time version of (2), that is

$$\mathbf{x}_{t+\delta t} = \mathbf{x}_t + \frac{\delta t}{2} \nabla \log \pi(\mathbf{x}_t) + \delta t \mathbf{z}_t, \qquad \mathbf{z}_t \sim N(\mathbf{0}, \mathbf{I}),$$

and we linearize the gradient around $\mathbb{E}(\mathbf{x}_t)$, we obtain

$$\mathbb{E}(\mathbf{x}_{t+\delta t}) \approx \mathbb{E}(\mathbf{x}_t) + \frac{\delta t}{2} \nabla \log \pi\big\{\mathbb{E}(\mathbf{x}_t)\big\}, \tag{3}$$

and

$$\begin{aligned} \mathrm{Cov}(\mathbf{x}_{t+\delta t}) \approx & \left[\mathbf{I} + \frac{\delta t}{2} \nabla^2 \log \pi\big\{\mathbb{E}(\mathbf{x}_t)\big\}\right] \mathrm{Cov}(\mathbf{x}_t) \\ & \times \left[\mathbf{I} + \frac{\delta t}{2} \nabla^2 \log \pi\big\{\mathbb{E}(\mathbf{x}_t)\big\}\right]^T + \delta t \mathbf{I}, \end{aligned} \tag{4}$$

where $\nabla^2 \log \pi(\mathbf{x})$ is the Hessian of the target log-density and $\mathbf{I}$ is a d-dimensional identity matrix. In continuous-time this leads to the following differential equations

$$\dot{\boldsymbol{\mu}}_t = \frac{d\boldsymbol{\mu}_t}{dt} = \frac{1}{2} \nabla \log \pi(\boldsymbol{\mu}_t), \tag{5}$$

$$\begin{aligned} \dot{\boldsymbol{\Sigma}}_t &= \frac{d\boldsymbol{\Sigma}_t}{dt} \\ &= \left\{\frac{1}{2} \nabla^2 \log \pi(\boldsymbol{\mu}_t)\right\} \boldsymbol{\Sigma}_t + \boldsymbol{\Sigma}_t \left\{\frac{1}{2} \nabla^2 \log \pi(\boldsymbol{\mu}_t)\right\} + \mathbf{I}, \end{aligned} \tag{6}$$

where we defined $\boldsymbol{\mu}_t = \mathbb{E}(\mathbf{x}_t)$ and $\boldsymbol{\Sigma}_t = \mathrm{Cov}(\mathbf{x}_t)$. For details regarding how (3) and (4) lead to (5) and (6), see the Supplementary Material. Notice that if the gradient is linear, that is if $\nabla \log \pi(\mathbf{x}) = \mathbf{F}\mathbf{x}$ for some matrix $\mathbf{F}$, (5) and (6) are equivalent to the differential equations used to propagate the mean and covariance of the state process in the Kalman-Bucy filter (Bucy and Joseph, 1987), under the special circumstance that the observation and control processes are absent.

If $\pi(\mathbf{x})$ is Gaussian then $\nabla \log \pi(\mathbf{x})$ is linear and, given any initial state $\mathbf{x}_{t_0}$, (5) and (6) can be solved analytically. In addition, $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ will converge, as $t \to \infty$, to the mean vector and covariance matrix of $\mathbf{x}$ under $\pi(\mathbf{x})$. Hence, given that a Gaussian distribution is fully specified by its first two moments, a Gaussian target is recovered exactly. However, if the target is not Gaussian, several issues arise. Firstly (5) and (6) generally do not have analytic solutions. This is a relatively mild problem, which can be addressed by using a numerical integrator, such as a Runge-Kutta method

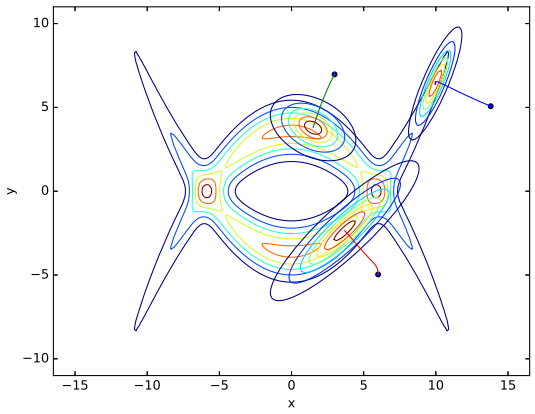

Fig. 1 Three local Gaussian approximations to a multimodal target density. The mean vectors and covariance matrices of the local densities were generated by solving (5) and (6), with $\boldsymbol{\mu}_{t_0}$ indicated by the three black dots.

(Ascher and Petzold, 1998). More importantly, the solutions to (5) and (6) will generally not converge to the true mean and covariance under $\pi(\mathbf{x})$, even as $t \to \infty$. To see this, assume that $\pi(\mathbf{x})$ is unimodal. Given that the solution to (5) is a steepest ascent curve, $\boldsymbol{\mu}_t$ will eventually converge to the mode of $\pi(\mathbf{x})$. However, unless $\pi(\mathbf{x})$ is symmetric, its mode differs from its mean vector.

The second issue entails that, unless $\nabla \log \pi(\mathbf{x})$ is linear, the quality of the approximation to the first two moments will typically degrade as $t - t_0$ increases, regardless of the numerical integrator used. This is not of great concern in our case, because we are interested in creating local, not global, approximations to $\pi(\mathbf{x})$. In particular, let $p(\mathbf{x}_{t_1}|\mathbf{x}_{t_0})$ be the distribution of $\mathbf{x}_{t_1}$, generated by integrating (2) between t_0 and a finite pseudo-time $t_1 > t_0$. Also, let $q(\mathbf{x}_{t_1}|\mathbf{x}_{t_0})$ be a Gaussian approximation to $p(\mathbf{x}_{t_1}|\mathbf{x}_{t_0})$, with mean and covariance matrix derived by solving (5) and (6), with the initial conditions $\boldsymbol{\mu}_{t_0} = \mathbf{x}_{t_0}$ and $\boldsymbol{\Sigma}_{t_0} = \delta t \mathbf{I}$. Our proposal is based on the observation that, while $q(\mathbf{x}_{t_1}|\mathbf{x}_{t_0})$ might not represent a good global approximation to $\pi(\mathbf{x})$ for any value of t_1 or $\mathbf{x}_{t_0}$, it often provides an accurate Gaussian approximation to the distribution of random variables generated from $\pi(\mathbf{x})$, in the vicinity of $\boldsymbol{\mu}_{t_1}$. As an example, consider the highly non-Gaussian density represented in Figure 1. In addition to the target, we show three Gaussian densities, obtained by numerically integrating (5) and (6) from three starting points, using 100 steps and $\delta t = 0.04$. Notice how the covariance matrices adapt to the local shape of the target.

In the context of importance sampling, an accurate global approximation to $\pi(\mathbf{x})$ is needed. We propose to create such a density using a mixture of local Gaussian approximations $q(\mathbf{x}_{t_1}|\mathbf{x}^1_{t_0}), \ldots, q(\mathbf{x}_{t_1}|\mathbf{x}^k_{t_0})$. The IMIS algorithm provides a natural approach to determining the initial positions, $\mathbf{x}^1_{t_0}, \ldots, \mathbf{x}^k_{t_0}$, because it places additional mixture components where the importance density is lacking mass, relative to the target. To use the new local linearization within IMIS, it is sufficient to modify step 2(a) of Algorithm 1 as follows:

2(a)* Let $\mathbf{x}_j$ be the sample with the largest weight. Given the initial position $\boldsymbol{\mu}_{t_0} = \mathbf{x}_j$, covariance matrix $\boldsymbol{\Sigma}_{t_0} = \delta t \mathbf{I}$ and a user-defined pseudo-time t_1, obtain the approximate solutions, $\hat{\boldsymbol{\mu}}_{t_1}$ and $\hat{\boldsymbol{\Sigma}}_{t_1}$, by numerically integrating (5) and (6). Then, set $\boldsymbol{\mu}_k = \hat{\boldsymbol{\mu}}_{t_1}$ and $\boldsymbol{\Sigma}_k = \hat{\boldsymbol{\Sigma}}_{t_1}$ and proceed to step 2(b).

We refer to this modified version of Algorithm 1 as Langevin Incremental Mixture Importance Sampling (LIMIS). In our experience, the choice of $\boldsymbol{\Sigma}_{t_0}$ is not particularly critical, as long as this matrix is positive semi-definite and on the scale of δt. However, if the target is log-concave, one might consider the less general initialization $\boldsymbol{\Sigma}_{t_0} = -\delta t \{\nabla^2 \log \pi(\boldsymbol{\mu}_{t_0})\}^{-1}$. Note that for all practical purposes we can assume that $t_0 = 0$, so the user needs specify only the final time t_1. This can be done manually or using the automated approach described in Section 7.

LIMIS has several advantageous properties. Firstly, $\hat{\boldsymbol{\Sigma}}_{t_1}$ is guaranteed to be positive definite, even when $\pi(\mathbf{x})$ is not log-concave. This is easily seen by considering discrete-time case, and noticing that the r.h.s. of (4) is positive definite. Secondly, the resulting approximation does not use a non-parametric estimator, such as Nearest Neighbour, to determine $\boldsymbol{\Sigma}_k$. As will be shown in Section 5 this is especially advantageous in high dimensions. Thirdly, as t_1 increases, the mixture components move toward the nearest mode of $\pi(\mathbf{x})$. This feature has been found to be advantageous by West (1992), who noticed that mixture approximations are typically over-dispersed relative to the target density, and proposed to shrink the mixture components towards the sample mean. Our experience suggests that shrinkage often leads to substantial efficiency gains, especially in high dimensions, but, as noted by Givens and Raftery (1996), West's method is less appropriate when the target is highly non-Gaussian, as in Figure 1, because it might shift some of the mixture densities toward areas where the target density is very low (such as the center of the plot). This cannot occur when (5) is used, because $\boldsymbol{\mu}_t$ moves toward areas where the target density is strictly higher. Finally, the most important property of LIMIS is that it provides an extremely parsimonious parametrization of the mixture importance density. Indeed, the locations and covariances of the mixture components are determined by equations (5)

and (6). Through these, LIMIS extracts local information about the target, which allows it to limit the number of free parameters that determine the shape of the mixture density to one: the final pseudo-time t_1.

4 Step-size selection

As explained in Section 3, equations (5) and (6) can be used to propagate the mean vector, $\boldsymbol{\mu}_t$, and covariance matrix, $\boldsymbol{\Sigma}_t$, of each mixture component between t_0 and t_1. In general, the solutions will be approximated using a numerical integrator, such as a Runge-Kutta scheme. Given a target density $\pi(\mathbf{x})$, let $L^{\pi}_{\boldsymbol{\mu}}(\boldsymbol{\mu}, \delta t)$ and $L^{\pi}_{\boldsymbol{\Sigma}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \delta t)$ be the operators used to update the moments, that is

$$\hat{\boldsymbol{\mu}}_{t+\delta t} = L^{\pi}_{\boldsymbol{\mu}}(\boldsymbol{\mu}_t, \delta t), \quad \hat{\boldsymbol{\Sigma}}_{t+\delta t} = L^{\pi}_{\boldsymbol{\Sigma}}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t, \delta t).$$

These operators depend on the numerical scheme used. For instance, if an Euler scheme is used, they are given by the r.h.s. of (3) and (4) (after discarding the $O(\delta t^2)$ term in (4)). If $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ represent the true solutions of (5) and (6), then the local truncation errors of the numerical integrator are

$$\boldsymbol{e}_{\boldsymbol{\mu}} = \boldsymbol{\mu}_{t+\delta t} - \hat{\boldsymbol{\mu}}_{t+\delta t}, \quad \boldsymbol{e}_{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{t+\delta t} - \hat{\boldsymbol{\Sigma}}_{t+\delta t},$$

which are generally $O\{(\delta t)^{\psi}\}$, for $\psi > 1$ (Süli and Mayers, 2003). While it is possible to choose δt so that numerical estimates of $|\boldsymbol{e}_{\boldsymbol{\mu}}|$ and $|\boldsymbol{e}_{\boldsymbol{\Sigma}}|$ are below certain thresholds, here we propose a different approach. In particular, we describe a novel statistically-motivated measure of discretization quality, which we then use to determine the step-size δt.

Our proposal consists in quantifying the integration quality in terms of distance between two local Gaussian densities: $q(\mathbf{x}) = \phi(\mathbf{x}|\hat{\boldsymbol{\mu}}_{t+\delta t}, \hat{\boldsymbol{\Sigma}}_{t+\delta t})$ and $q^*(\mathbf{x}) = \phi(\mathbf{x}|\boldsymbol{\mu}_{t+\delta t}, \boldsymbol{\Sigma}_{t+\delta t})$. While there are other distance measures that could be adopted, such as the Kullback-Leibler (KL) divergence, we would like a measure that is easily interpretable. For this reason we consider the Population Effective Sample Size (PESS), which we define as

$$\begin{aligned} \mathrm{PESS}\big\{q(\mathbf{x}), q^*(\mathbf{x})\big\} &= \operatorname*{plim}_{n\to\infty} \frac{\mathrm{ESS}^{\mathrm{IS}}\big\{q(\mathbf{x}), q^*(\mathbf{x})\big\}}{n} \\ &= \left[\int \left\{\frac{q(\mathbf{x})}{q^*(\mathbf{x})}\right\}^2 q^*(\mathbf{x}) d\mathbf{x}\right]^{-1}, \end{aligned} \tag{7}$$

where

$$\mathrm{ESS}^{\mathrm{IS}}\big\{q(\mathbf{x}), q^*(\mathbf{x})\big\} = \left\{\sum_{i=1}^{n} \frac{q(\mathbf{x}_i)}{q^*(\mathbf{x}_i)}\right\}^2 \Big/ \sum_{i=1}^{n} \left\{\frac{q(\mathbf{x}_i)}{q^*(\mathbf{x}_i)}\right\}^2, \tag{8}$$

is the Effective Sample Size (ESS) measure proposed by Kong et al (1994) and $\mathbf{x}_i \sim q^*(\mathbf{x})$ for $i = 1, \ldots, n$. As we show in the Supplementary Material, when both $q(\mathbf{x})$ and $q^*(\mathbf{x})$ are Gaussian densities, it is possible to obtain an analytic expression for the PESS

$$\begin{aligned} \mathrm{PESS}\big\{q(\mathbf{x}), q^*(\mathbf{x})\big\} = \Bigg[&\left(|2\boldsymbol{\Sigma}_{q^*} - \boldsymbol{\Sigma}_q|\right)^{-\frac{1}{2}} |\boldsymbol{\Sigma}_q|^{-\frac{1}{2}} |\boldsymbol{\Sigma}_{q^*}| \\ &\times \exp\left\{(\boldsymbol{\mu}_{q^*} - \boldsymbol{\mu}_q)^T (2\boldsymbol{\Sigma}_{q^*} - \boldsymbol{\Sigma}_q)^{-1} (\boldsymbol{\mu}_{q^*} - \boldsymbol{\mu}_q)\right\}\Bigg]^{-1}. \end{aligned} \tag{9}$$

This distance measure has the advantage of having a clear statistical interpretation: it is the limiting value of the ESS, normalized by the number of samples n. Recall that we are solving (5) and (6) in order to construct an additional density to be added to the importance mixture. Hence, at each step of the numerical integrator, we are not interested in assessing the accuracy of the approximate solutions $(\hat{\boldsymbol{\mu}}_{t+\delta t}, \hat{\boldsymbol{\Sigma}}_{t+\delta t})$ per se, but we want to quantify how the discretization error perturbs the corresponding density, $q(\mathbf{x})$, away from $q^*(\mathbf{x})$. Therefore, we prefer using (9), rather the truncation errors $\boldsymbol{e}_{\boldsymbol{\mu}}$ and $\boldsymbol{e}_{\boldsymbol{\Sigma}}$, to determine the steps size. Notice also that $\mathrm{PESS}\big\{q(\mathbf{x}), q^*(\mathbf{x})\big\} \in [0, 1]$, as long as $2\boldsymbol{\Sigma}_{q^*} - \boldsymbol{\Sigma}_q$ is positive definite, while $\mathrm{KL}\big\{q(\mathbf{x}), q^*(\mathbf{x})\big\} \geq 0$. Most importantly, by looking at (7) it is simple to realize that the chosen criterion is invariant under transformation of $\mathbf{x}$, which certainly is not the case for $|\mathbf{e}_{\boldsymbol{\mu}}|$ and $|\mathbf{e}_{\boldsymbol{\Sigma}}|$.

Having defined an appropriate distance measure, the step size δt can be selected at t_0, and kept fixed afterward, or adaptively at each step. Here we follow the former approach. In particular, if we indicate with $\boldsymbol{\mu}_{t_0}$ and $\boldsymbol{\Sigma}_{t_0}$ the initial moments, then the step-size is selected as follows

$$\begin{aligned} \delta t^* = \Big[\delta t \,:\, \mathrm{PESS}\big\{ \;& \phi(\mathbf{x}|\hat{\boldsymbol{\mu}}_{t_0+\delta t}, \hat{\boldsymbol{\Sigma}}_{t_0+\delta t}), \\ & \phi(\mathbf{x}|\boldsymbol{\mu}_{t_0+\delta t}, \boldsymbol{\Sigma}_{t_0+\delta t}) \big\} = \alpha\Big], \end{aligned} \tag{10}$$

where the parameter $\alpha \in (0, 1)$ is user-defined. Increasing α reduces the step-size, which leads to more accurate, but computationally more expensive, solutions to (5) and (6). We use $\alpha = 0.99$ as default value. While the true moments $(\boldsymbol{\mu}_{t+\delta t}, \boldsymbol{\Sigma}_{t+\delta t})$, needed to compute (10), are typically unknown, they can be approximated by propagating the moments between t_0 and $t_0 + \delta t$ using, say, 10 steps of size $\delta t/10$. Finally, (10) can generally be solved in just a few iterations by a standard one-dimensional root-finding algorithm, such as Brent's method (Brent, 2013), hence the cost of tuning δt is typically a small fraction of the overall computational cost of LIMIS.

5 Examples

Here we compare the new LIMIS sampler with NIMIS, IS and MALA on three examples. The first is a multimodal mixture density, whose components are warped Gaussian densities, defined on up to 80 dimensions. In the second we sample the posterior of a Bayesian logistic regression model, using the Sonar dataset of Gorman and Sejnowski (1988). The final example is the the ridge-like posterior density used by Raftery and Bao (2010) to test the original IMIS algorithm.

5.1 Set-up

We compare the performance of the samplers using several criteria. While some of these, such as Root Mean Squared Errors (RMSEs), are well known, others are less well known and so specified here. In Section 5.2 we evaluate the methods using the marginal accuracy measure of Faes et al (2011), that is

$$\text{MA} = 1 - \frac{1}{2}\int |\pi(\mathbf{x}) - \hat{\pi}(\mathbf{x})| d\mathbf{x},$$

where $\text{MA} = 1$ if $\pi(\mathbf{x})$ and $\hat{\pi}(\mathbf{x})$ are identical and $\text{MA} = 0$ if the two densities do not overlap anywhere. When weighted samples $\mathbf{z}_1, \ldots, \mathbf{z}_n$, drawn from $q(\mathbf{z})$, are available, $\pi(\mathbf{x})$ is estimated by

$$\hat{\pi}(\mathbf{x}) = \frac{1}{hn}\sum_{i=1}^{n} \kappa_h(\mathbf{x}|\mathbf{z}_i) w_i \approx \int \kappa_h(\mathbf{x}|\mathbf{z}) \frac{\pi(\mathbf{z})}{q(\mathbf{z})} q(\mathbf{z}) d\mathbf{z},$$

with $\kappa_h(\mathbf{x}|\mathbf{z})$ being a kernel density, with bandwidth h. An additional criterion is efficiency (EF), by which we indicate the ratio of ESS to total number of samples n. For LIMIS, NIMIS and IS we use formula (8) to compute the ESS, while for MALA we use

$$\text{ESS}^{\text{MC}} = \frac{n}{1 + 2\sum_{t=1}^{\infty} \rho_t},$$

where ρ_t is the autocorrelation of the chain at lag t. Notice that under both definitions $\text{ESS} \in [1, n]$, so $\text{EF} \in [0, 1]$. We indicate with EF_c the efficiency of a sampler divided by its total running time, in seconds.

We report the RMSEs of the estimated marginal means, variances and normalizing constant of the target ($\int \pi(\mathbf{x}) d\mathbf{x}$). While estimating the normalizing constant is straightforward when importance samples are available, much more care is required when using MCMC methods. Hence, we do not estimate this quantity when applying MALA.

In terms of algorithmic parameters, for IS, LIMIS and NIMIS we use $\nu = 3$, which is the smallest integer value of ν such that the variance of a Student's t random variable is finite, and we follow Raftery and Bao (2010) who suggest the default values $n_0 = 1000d$, $b = 100d$. We use an equal number $(n_0 + kb)$ of samples or iterations for IS and MALA. The step size of LIMIS is determined as explained in Section 4. The only LIMIS parameter that we chose manually is the final pseudo-time t_1. However, we discuss how it can be selected in an automated fashion in Section 7. When applying MALA we discard the first tenth of each MCMC chain as the burn-in period, and we select the step size so as to approximately achieve the optimal 0.574 acceptance rate derived by Roberts et al (2001). The remaining settings will be detailed in Sections 5.2 and 5.3.

All the examples are implemented in the `Julia` language (Bezanson et al, 2012). We developed our own implementation of LIMIS and NIMIS, while we use the MALA algorithm offered by the `Klara` `Julia` package.

5.2 Mixture of warped Gaussians

As a first example we consider a mixture target density

$$\pi(\mathbf{x}) = \sum_{i=1}^{r} w_i p_i(\mathbf{x}), \quad \sum_{i=1}^{r} w_i = 1,$$

where each of the r mixture components is a shifted version of the banana-shaped density described in Haario et al (2001). In particular, let $\mathbf{y} \sim N(0, \boldsymbol{\Sigma}_a)$, where $\boldsymbol{\Sigma}_a = \text{diag}(a^2, 1, \ldots, 1)$, and consider the following transformed random variables

$$x_1 = y_1 + s_1, \quad x_2 = y_2 - b(y_1^2 - a^2) + s_2, \quad x_i = y_i,$$

for $i = 3, \ldots, d$, and where a, b, s_1 and s_2 are constants. Given that the determinant of the Jacobian of this transformation is 1, the density of $\mathbf{x}$ is simply

$$p(\mathbf{x}) = \phi[x_1 - s_1, x_2 + b\{(x_1 - s_1)^2 - a^2\} - s_2, x_3 \ldots, x_d | \mathbf{0}, \boldsymbol{\Sigma}_a],$$

where $\phi(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the p.d.f. of a multivariate normal distribution and $\mathbf{0}$ is a d-dimensional vector of zeros. We consider a mixture of $r = 6$ such densities, each with different values for parameters a, b, s_1 and s_2. These are reported in the Supplementary Material, together with formulas for the gradient and Hessian of $\log \pi(\mathbf{x})$. A slice of the target density across the first two dimensions is shown in top-left plot of Figure 2.

We sample $\pi(\mathbf{x})$ using LIMIS, NIMIS, IS and MALA. In particular, we consider three scenarios where d is respectively equal to 5, 20 and 80. For LIMIS and NIMIS we use $k = 200$ iterations, and for the former method we let t_1 grow with d, by setting it to 1, 3 and 5. To make sure that the initial sample covers all regions of high target density, which is particularly critical in high

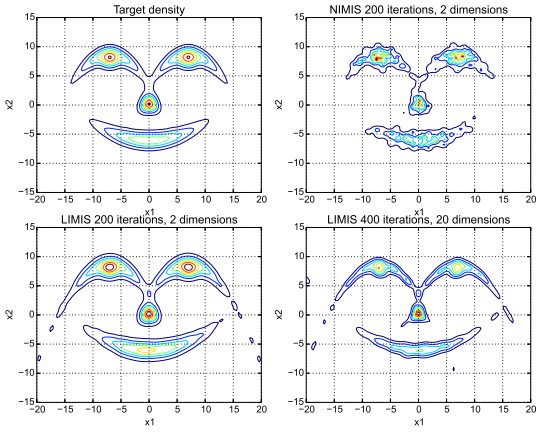

Fig. 2 Two-dimensional slice of the target and of the importance densities, obtained using LIMIS and NIMIS.

dimensions, we initialize LIMIS and NIMIS using a diffuse $\text{mvt}(\mathbf{x}|\mathbf{0}, 100\mathbf{I}, 3)$ prior distribution. To perform IS we use a weighted mixture of four multivariate Student's t distributions each centered at one of the modes of the target, $\mathbf{x}_1^*, \ldots, \mathbf{x}_4^*$, and with covariances equal to $-2\{\nabla^2 \log \pi(\mathbf{x})\}|^{-1}_{\mathbf{x}=\mathbf{x}_i^*}$, for $i = 1, \ldots, 4$. The weights are reported in the Supplementary Material. See Section 5.1 for additional information about the simulation settings.

Table 1 reports the results obtained using 16 independent runs of each sampler. We do not report the efficiency of MALA, because the ESS was extremely low in all runs, even when the algorithm was mixing properly. In fact, sample autocorrelations are high when the sampler explores the same mode for several hundred iterations before jumping to another mode, which results in extremely low autocorrelation-adjusted ESS estimates. Hence, we consider the resulting efficiency estimates to be misleading.

In the five-dimensional scenario NIMIS closely follows LIMIS, which is the best performer on most criteria. However, the performance of NIMIS degrades rapidly as the dimensionality increases, to the point that it failed entirely in 80 dimensions. IS achieves the highest EF_c in five dimensions, which is not surprising given that the target density is cheap to evaluate and that the importance density was chosen manually. However, IS loses its efficiency advantage in higher dimensions. LIMIS seems to be scaling well with d on most criteria, the estimated marginal variances, $\sum_{i=3}^{d} \text{Var}(x_i)$, being an exception. Here MALA achieves a lower MSE than LIMIS when $d = 20$, and the gap increases when $d = 80$. However, in 80 dimensions, LIMIS is still more accurate than MALA in terms of marginal accuracies and of the RMSE for $\sum_{i=3}^{d} \mathbb{E}(x_i)$.

5.3 Logistic Regression

Here we consider a Bayesian logistic regression problem. Assume we have n i.i.d samples of binary labels $\mathbf{y} \in \{0, 1\}^n$ and a corresponding $n \times d$ matrix of covariates $\mathbf{X}$. Under a logistic regression model

$$\text{Prob}(y_i = 1|\mathbf{X}, \boldsymbol{\theta}) = \frac{e^{\mathbf{X}_{i:}^T \boldsymbol{\theta}}}{1 + e^{\mathbf{X}_{i:}^T \boldsymbol{\theta}}}, \quad \text{for } i = 1, \ldots, n,$$

where $\boldsymbol{\theta}$ is a vector of model coefficients and $\mathbf{X}_{i:}$ is a column vector giving the i-th row of $\mathbf{X}$. If $X_{j1} = 1$, for $j = 1, \ldots, n$, then θ_1 represents the intercept. If we use a flat prior on θ_1 and a Gaussian prior on $\{\theta_2, \ldots, \theta_d\}$, with mean zero and covariance $\mathbf{I}\lambda^{-1}$, where $\mathbf{I}$ is a $d-1$ dimensional identity matrix and $\lambda > 0$, the posterior log-density of the parameters is

$$\log \pi(\boldsymbol{\theta}) \propto \mathbf{y}^T \mathbf{X}\boldsymbol{\theta} - \sum_{i=1}^{n} \log(1 + e^{\mathbf{X}_{i:}^T \boldsymbol{\theta}}) - \frac{\lambda}{2} \sum_{j=2}^{d} \theta_j^2.$$

Formulas for the gradient and Hessian of $\log \pi(\boldsymbol{\theta})$ are provided in the Supplementary Material.

To verify how LIMIS performs on this model, we consider the Sonar dataset, which is freely available within the UCI repository (Lichman, 2013). The dataset was originally considered by Gorman and Sejnowski (1988), who used it to train a neural network to discriminate sonar signals bounced off a mine from those bounced off a rock. It includes $n = 208$ observations, where the response variable indicates whether the object is a mine ($y = 1$) or a rock ($y = 0$). Each covariate vector contains $d = 60$ numbers ranging between 0 and 1, which represent the signal's energy within a specific frequency interval, integrated over time. See Gorman and Sejnowski (1988) for more details on the dataset.

We aim at sampling $\pi(\boldsymbol{\theta})$ using LIMIS, NIMIS, IS and MALA, for fixed λ. We consider two different scenarios. In the first, after standardizing the features $\mathbf{X}$, we select $\lambda \approx 28$ by k-fold cross-validation. In the second, we use a much weaker penalization by choosing $\lambda = 1$. For LIMIS and NIMIS we use $k = 100$ iterations, and for MALA we discard the first 10% of each chain as burn-in period. As importance distribution for IS we use a multivariate Student's t distribution with 3 degrees of freedom, centred at the posterior mode, $\boldsymbol{\theta}^*$, and with covariance matrix equal to $-2\{\nabla^2 \log \pi(\boldsymbol{\theta})\}|^{-1}_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}$. We use the same density to initialize LIMIS and NIMIS. The remaining settings are as described in Section 5.1.

Table 2 summarizes the results, on both scenarios, of 16 independent estimation runs. The first three rows report the RMSEs of the estimated marginal posterior means and variances, averaged over the 61 dimensions, and of the estimated normalizing constant or marginal

5d	LIMIS	NIMIS	IS	MALA	ord. mag.
$\text{MA}(x_1)$	**0.991**	0.989	0.965	0.954	1
$\text{MA}(x_2)$	**0.990**	0.988	0.967	0.929	1
$\sum_{i=3}^{d} \mathbb{E}(x_i)$	**0.53**(0.99)	0.62(0.98)	1.74(0.98)	0.84(0.91)	10^{-2}
$\sum_{i=3}^{d} \text{Var}(x_i)$	**9.11**(0.97)	15.96(0.42)	33.12(0.97)	10.91(0.85)	10^{-3}
$\int \pi(\mathbf{x}) d\mathbf{x}$	**2.30**(0.35)	5.80(0.17)	13.26(0.99)	-	10^{-3}
$\text{EF}(\text{EF}_c)$	**0.69**(0.094)	0.52(0.084)	0.05(**0.106**)	-	1(1)
20d					
$\text{MA}(x_1)$	**0.994**	0.846	0.942	0.970	1
$\text{MA}(x_2)$	**0.993**	0.844	0.943	0.966	1
$\sum_{i=3}^{d} \mathbb{E}(x_i)$	**0.97**(0.99)	8.58(0.73)	11.56(0.94)	1.54(0.98)	10^{-2}
$\sum_{i=3}^{d} \text{Var}(x_i)$	47.73(0.29)	4429(0.01)	771(0.94)	**20.67**(0.85)	10^{-3}
$\int \pi(\mathbf{x}) d\mathbf{x}$	**2.45**(0.25)	306.4(0.01)	57.56(0.98)	-	10^{-3}
$\text{EF}(\text{EF}_c)$	**0.416**(**0.77**)	0.005(0.006)	0.008(0.23)	-	$1(10^{-2})$
80d					
$\text{MA}(x_1)$	**0.995**	-	0.945	0.982	1
$\text{MA}(x_2)$	**0.995**	-	0.947	0.980	1
$\sum_{i=3}^{d} \mathbb{E}(x_i)$	**1.13**(0.88)	-	32.14(0.86)	2.8(0.96)	10^{-2}
$\sum_{i=3}^{d} \text{Var}(x_i)$	113.2(0.35)	-	3029(0.37)	**27.2**(0.99)	10^{-3}
$\int \pi(\mathbf{x}) d\mathbf{x}$	**1.6**(0.37)	-	41.1(0.50)	-	10^{-3}
$\text{EF}(\text{EF}_c)$	**0.22**(**0.404**)	-	0.002(0.044)	-	$1(10^{-3})$

Table 1 Results for mixture of warped Gaussians, for each dimension: a) the first two rows report marginal accuracies along the first two dimensions; b) the following three rows contain RMSEs and, between brackets, the ratio between squared bias and MSE; c) the last row reports mean efficiencies and, between brackets, mean corrected efficiencies. For each row, the order of magnitude of the marginal accuracies, RMSEs and efficiencies is reported in the last column.

$\lambda \approx 28$	LIMIS	NIMIS	IS	MALA	scale
$\mathbb{E}(\theta_j)$	**4.7**(0.95)	22.5(0.87)	5.9(0.94)	11.1(0.95)	10^{-4}
$\text{Var}(\theta_j)^{1/2}$	**3.3**(0.94)	12.9(0.91)	4.1(0.94)	5.9(0.92)	10^{-4}
$\int \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}$	**2.3**(0.97)	50.0(0.05)	3.3(0.89)	-	10^{-3}
$\text{EF}(\text{EF}_c)$	**0.18**(1.15)	0.01(0.04)	0.11(**3.55**)	0.03(0.50)	$1(10^{-3})$
$\lambda = 1$					
$\mathbb{E}(\theta_j)$	**94.7**(0.92)	393.2(0.91)	221.9(0.95)	148.2(0.95)	10^{-4}
$\text{Var}(\theta_j)^{1/2}$	**46.4**(0.90)	151.2(0.90)	147.1(0.94)	64.4(0.92)	10^{-4}
$\int \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}$	118.8(0.01)	668.9(0.01)	**37.4**(0.93)	-	10^{-3}
$\text{EF}(\text{EF}_c)$	**0.015**(**1.2**)	0.0019(0.07)	0.0014(0.47)	0.0038(0.62)	$1(10^{-4})$

Table 2 Results for both logistic regression scenarios, first three rows: RMSE and, between brackets, the ratio between squared bias and MSE, for each estimate and method. Last row: mean efficiency and, between brackets, mean corrected efficiency. Last column: order of magnitude of RMSEs and efficiencies.

likelihood. When $\lambda \approx 28$, LIMIS is the best method terms of RME and EF, but IS achieves the lowest EF_c. IS does well because this value of λ results in an approximately Gaussian posterior. All methods perform less well when the prior is more dispersed, and the posterior farther from Gaussian. This is the case also for MALA, even though this method does not rely on a global approximation to the posterior. When $\lambda = 1$ LIMIS is more efficient the IS, in terms of both EF and EF_c. The fact that EF is ten times higher under LIMIS than under IS implies that producing an equivalent number of effective sample requires ten times more storage under IS. This is important in parallel environments where memory bandwidth, rather than computational resources, are often the main bottleneck. As expected, NIMIS is the worst performing method in both scenarios, due to the high dimensionality of the target.

5.4 Ridge-like density

Here we consider a model, originally proposed by Bates (2001), whose posterior density lies along a very thin ridge. The model has six parameters, while prior and likelihood densities are

$$p(\boldsymbol{\theta}) = \prod_{i=1}^{6} \phi(\theta_i | \gamma_i, \beta_i^2), \quad p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{i=1}^{4} \phi(y_i | \mu_i, \sigma_i^2),$$

where $\mu_1 = \prod_{i=1}^{6} \theta_i$, $\mu_2 = \theta_2\theta_4$, $\mu_3 = \theta_1/\theta_5$ and $\mu_4 = \theta_3\theta_6$. The values of the γs, βs, ys and σs are reported in the Supplementary Material, which contains also expressions for gradient and Hessian of the log-posterior.

Notice that the μs and the ys are exchanged here, relative to Raftery and Bao (2010).

We sample the posterior using the following setup. The moments of the Gaussian importance density used by IS are determined as in Section 5.4. We use $k = 500$ iterations LIMIS and NIMIS, and for the latter we set $t_1 = 0.001$. We choose a small integration interval because using the default value of α within the step-size selection approach of Section 4 leads to $\delta t = O(10^{-6})$. Given that the solutions to (5) are steepest ascent curves, it is not surprising that a discretized solution needs to use very small steps, at least in the vicinity of the ridge, in order go uphill. All remaining settings are described in Section 5.1.

Table 3 contains the results of 16 simulation runs. The first three rows report the RMSEs of the estimated marginal posterior means and variances, averaged over the six dimensions, and of the estimated marginal likelihood. The performance LIMIS and NIMIS is similar to that obtained in the five-dimensional warped Gaussian mixture example. In particular, both methods are able to concentrate the mass of the importance density along the non-linear ridge of the posterior. This cannot be achieved using a single Gaussian, and in fact IS performs poorly here. Surprisingly, MALA performs worse than IS on this example, despite its use of gradient information. It is possible that including second order information within MALA, as proposed by Girolami and Calderhead (2011), would lead to better mixing. However, this target density is not log-concave, hence it would be necessary to perturb the Hessian matrix in order obtain a positive definite scaling matrix.

6 Computational considerations

In the previous examples we reported the corrected efficiencies, EF_c, of the different samplers. Even though all methods were implemented in the same language, their relative performance is still strongly dependent on the simulation setting. For instance, let k be the number of iterations of NIMIS and LIMIS, and let b be the number of samples simulated at each step. Then, for fixed k, increasing b improves the efficiency of LIMIS relative to that of NIMIS. This is because the cost of integrating (5) and (6), which does not depend on b, is amortized over a larger number of samples, while NIMIS needs to search across a larger number of neighbours in step 2(a). Despite these intricacies, here we make some broad consideration about computational efficiency, which should be useful in a practical setting.

Let $n_k = n_0 + kb$ be the total number of samples obtained using LIMIS, NIMIS, MALA and IS. An important factor in determining the attractiveness of each method is the cost of evaluating $\log \pi(\mathbf{x})$ and its derivatives. MALA requires n_k evaluations of $\nabla \log \pi(\mathbf{x})$. LIMIS evaluates gradient and Hessian several times when constructing the k mixture densities. The factor multiplying k depends on the number of steps used in the Langevin linearization of Section 3. For instance, using the default α proposed in Section 4 to determine the step size, δt, the linearization requires on average 50 integration steps in the twenty-dimensional warped Gaussian mixture example. In that scenario we used $k = 200$, $b = 100d$ and $n_0 = 1000d$, hence a whole LIMIS run requires around 10^4 evaluations of gradient and Hessian, which should be compared with the $n_0 + kb = 42 \times 10^4$ gradient evaluations required by MALA. The Hessian of this example is highly sparse but, for a typical model, computing it should be $O(d)$ times more expensive than evaluating the gradient. Hence, if the Hessian of this example was dense, the total cost of computing the derivatives under LIMIS and MALA would roughly match. However, notice that LIMIS outputs a mixture density which can be used to do further importance sampling, and this does not require any additional derivative evaluation.

A second factor is the cost of evaluating the importance density. At the j-th iteration of LIMIS or NIMIS, where $j \in \{1, \ldots, k\}$, the cost of single evaluation is $O(jd^2)$. While in the examples we ran these algorithms until a fixed k was reached, it might be preferable to stop when the increased cost of evaluating the importance mixture is not more than offset by gains in efficiency. In particular, let c_π and c_q be, respectively, the cost of evaluating the density of the target or of a single mixture component. Then the cost of an independent sample is approximately

$$c(j) = \frac{c_\pi + jc_q}{\mathrm{EF}(j)}, \quad \text{for } j = 1, 2, \ldots, \tag{11}$$

where $\mathrm{EF}(j)$ is the efficiency of an importance mixture with j components. Figure 3 shows the behaviour of $c(j)$ when running LIMIS on the mixture example. For more complex examples accurate time estimates would be required, but here the target is a mixture of six warped Gaussian densities, hence we assumed $c_\pi \approx 6c_q$. The plot suggests that the computational budget could be used more efficiently by stopping LIMIS around the 25th iteration, and using the resulting mixture importance density to obtain more samples.

In the previous examples we have seen that, from the point of view of statistical efficiency, the performance of NIMIS is very unsatisfactory in high dimensions. IS scales better, as long as a good approximation to the target is available, as in the logistic example. IS is also computationally cheap, because it does not require

	LIMIS	NIMIS	IS	MALA	scale
$\mathbb{E}(\theta_j)$	**0.18**(0.91)	0.32(1.00)	2.68(0.99)	44.6(0.96)	10^{-3}
$\mathrm{Var}(\theta_j)^{1/2}$	**0.22**(0.72)	0.23(0.80)	3.36(0.89)	20.5(0.87)	10^{-3}
$\int \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}$	**0.32**(0.07)	0.62(0.10)	2.15(0.61)	-	10^{-2}
$\mathrm{EF}(\mathrm{EF}_c)$	**0.53**(**0.43**)	0.39(0.40)	0.013(0.06)	$0.0001(< 10^{-3})$	$1(10^{-2})$

Table 3 Results for ridge-like model, first three rows: RMSE and, between brackets, the ratio between squared bias and MSE, for each estimate and method. Last row: mean efficiencies and, between brackets, mean corrected efficiencies on the 16 runs. Last column: order of magnitude of RMSEs and efficiencies.

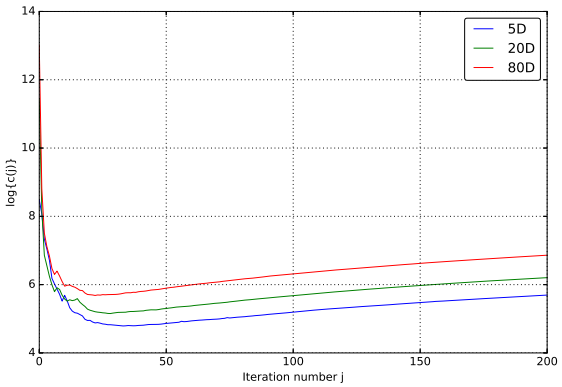

Fig. 3 Log-cost per sample, averaged over 16 runs, under the three scenarios considered in Section 5.2.

derivative information. However, in high-dimensional non-Gaussian scenarios a good off-the-shelf importance distribution is, in most cases, not readily available. In these cases, using LIMIS to construct an efficient importance distribution might be advantageous. In fact, the output mixture density could then be used to obtain more importance samples, which would not require any further evaluations of the target's gradient and Hessian. Obviously, one has to be careful not grow the size of the importance mixture to the point that the increase in the cost of evaluating this density is not justified by the resulting statistical efficiency gains. This could be avoided by stopping LIMIS when a criterion such as (11) is approximately minimized.

7 Tuning the final pseudo-time t_1

In the examples presented in Section 5 we selected the final pseudo-time t_1 manually. In general we start from the default $t_1 = 1$ and check whether perturbing t_1 drastically improves a performance measure, such as EF. In the logistic regression example the performance did not seem to depend much on t_1, hence we used its default value. In the mixture density example we increased t_1 with the dimensionality, d, of the target. Increasing t_1 inflates the covariance of the importance mixture components and it shrinks their locations towards the closest mode of the target. Given that distances increase with d, this is a desirable behaviour. An alternative approach would have been to increase the number of LIMIS iterations with d, while keeping t_1 constant.

In this section we show how an initial choice of t_1 can be improved, using a fully automated procedure. Assume that the results of a preliminary LIMIS run are available, and include a weighted sample $\mathbf{x}_1, \ldots, \mathbf{x}_{n_k}$, where $n_k = n_0 + kb$ and k is the number of LIMIS iterations. Indicate with $\boldsymbol{\mu}^1_{t_0} = \tilde{\mathbf{x}}_1, \ldots, \boldsymbol{\mu}^k_{t_0} = \tilde{\mathbf{x}}_k$ the samples that achieved the highest weight in one of the iterations, and hence resulted in the addition of a mixture component. Let $q(\mathbf{x}|t_1)$ be a Gaussian importance mixture, whose components have mean vectors, $\boldsymbol{\mu}^1_{t_1}, \ldots, \boldsymbol{\mu}^k_{t_1}$, and covariance matrices, $\boldsymbol{\Sigma}^1_{t_1}, \ldots, \boldsymbol{\Sigma}^k_{t_1}$, which are constructed by numerically integrating (5) and (6) between t_0 and t_1. In this section we aim at selecting t_1 so that $q(\mathbf{x}|t_1)$ is optimal, in a sense to be clarified shortly.

Suppose that we wish to estimate

$$I = \mathbb{E}\big\{h(\mathbf{x})\big\} = \int h(\mathbf{x}) \frac{\pi(\mathbf{x})}{c} d\mathbf{x} = \int h(\mathbf{x}) \tilde{\pi}(\mathbf{x}) d\mathbf{x},$$

where $c = \int \pi(\mathbf{x}) d\mathbf{x}$ and $h(\mathbf{x})$ is an $\mathbb{R}^d \to \mathbb{R}$ function. If only the un-normalized target, $\pi(\mathbf{x}) = c\,\tilde{\pi}(\mathbf{x})$, can be evaluated, then I can be estimated by self-normalized importance sampling, that is

$$\hat{I} = \frac{\sum_{j=1}^m h(\mathbf{x}_j) w_j}{\sum_{j=1}^m w_j}, \quad \text{where} \quad w_j = \frac{\pi(\mathbf{x}_j)}{q(\mathbf{x}_j|t_1)},$$

and $\mathbf{x}_j \sim q(\mathbf{x}_j|t_1)$, for $j = 1, \ldots, m$. The asymptotic variance of $\hat{I}$ is proportional to

$$v(t_1) = \frac{\int \frac{\pi(\mathbf{x})^2}{q(\mathbf{x}|t_1)^2} \big\{h(\mathbf{x}) - I\big\}^2 q(\mathbf{x}|t_1) d\mathbf{x}}{\big\{\int \frac{\pi(\mathbf{x})}{q(\mathbf{x}|t_1)} q(\mathbf{x}|t_1) d\mathbf{x}\big\}^2}, \tag{12}$$

hence, ideally, we would like to determine the value, t_1^*, that minimizes (12). In order to approximately achieve this, we need a reasonably cheap estimator of $v(t_1)$. Let $q(\mathbf{x}|t_1^I)$ be the mixture importance density in the final iteration of the pilot LIMIS run. Then (12) can be

estimated by

$$\hat{v}(t_1) = \frac{\frac{1}{n_k}\sum_{i=1}^{n_k} \frac{\pi(\mathbf{x}_i)^2}{q(\mathbf{x}_i|t_1)^2}\left\{h(\mathbf{x}_i) - \hat{I}\right\}^2 \frac{q(\mathbf{x}_i|t_1)}{q(\mathbf{x}_i|t_1^I)}}{\left\{\frac{1}{n_k}\sum_{i=1}^{n_k} \frac{\pi(\mathbf{x}_i)}{q(\mathbf{x}_i|t_1)}\frac{q(\mathbf{x}_i|t_1)}{q(\mathbf{x}_i|t_1^I)}\right\}^2} \quad (13)$$
$$= \frac{1}{\hat{c}^2 n_k}\sum_{i=1}^{n_k} \frac{\pi(\mathbf{x}_i)}{q(\mathbf{x}_i|t_1)}\left\{h(\mathbf{x}_i) - \hat{I}\right\}^2 w_i,$$

where

$$\hat{I} = \frac{1}{\hat{c}\, n_k}\sum_{i=1}^{n_k} h(\mathbf{x}_i) w_i, \quad \hat{c} = \frac{1}{n_k}\sum_{i=1}^{n_k} w_i, \quad w_i = \frac{\pi(\mathbf{x}_i)}{q(\mathbf{x}_i|t_1^I)}, \quad (14)$$

and $\mathbf{x}_i \sim q(\mathbf{x}_i|t_1^I)$, for $i = 1, \ldots, n_k$. Here $\mathbf{x}_i$, $h(\mathbf{x}_i)$, $\pi(\mathbf{x}_i)$, $q(\mathbf{x}_i|t_1^I)$, w_i, $\hat{I}$ and $\hat{c}$ have already been simulated/computed and stored during the preliminary run. Hence $\hat{v}(t_1)$ is a deterministic function, which can be minimized using a one-dimensional optimizer, where only $q(\mathbf{x}_i|t_1)$, for $i = 1, \ldots, n$, needs to be recomputed as the optimizer explores different values of t_1. If the normalized target, $\tilde{\pi}(\mathbf{x})$, can be computed directly and I is estimated using

$$\tilde{I} = \frac{1}{m}\sum_{j=1}^{m} h(\mathbf{x}_j)\frac{\tilde{\pi}(\mathbf{x}_j)}{q(\mathbf{x}_j|t_1)}, \quad \text{where} \quad \mathbf{x}_j \sim q(\mathbf{x}_j|t_1),$$

for $j = 1, \ldots, m$, then the finite-sample variance of $\tilde{I}$ is proportional to

$$\tilde{v}(t_1) = \int \frac{h(\mathbf{x})^2 \tilde{\pi}(\mathbf{x})^2}{q(\mathbf{x}|t_1)} d\mathbf{x}, \quad (15)$$

which can be estimated by

$$\hat{\tilde{v}}(t_1) = \frac{1}{n_k}\sum_{i=1}^{n_k} \frac{h(\mathbf{x}_i)^2 \tilde{\pi}(\mathbf{x}_i)}{q(\mathbf{x}_i|t_1)} w_i, \quad (16)$$

where

$$w_i = \frac{\tilde{\pi}(\mathbf{x}_i)}{q(\mathbf{x}|t_1^I)} \quad \text{and} \quad \mathbf{x}_i \sim q(\mathbf{x}|t_1^I),$$

for $i = 1, \ldots, n_k$. Also in this case only $q(\mathbf{x}_i|t_1)$ needs to be recomputed at t_1 varies.

Notice that, if we set $h(\mathbf{x}) = 1$ in (15), minimizing $\tilde{v}(t_1)$ is equivalent to maximizing $\mathrm{PESS}\{\tilde{\pi}(\mathbf{x}), q(\mathbf{x}|t_1)\}$ (7). This choice is useful when the practitioner is not interested in minimizing the variance under any particular integrand $h(\mathbf{x})$, but wants to obtain an importance density that is adapted to the target. However, in the self-normalized case, setting $h(\mathbf{x}) = 1$ leads to $v(t_1) = \hat{v}(t_1) = 0$ for any t_1, because this estimator is exact for constant $h(\mathbf{x})$. Hence, if the normalizing constant is unknown and no specific $h(\mathbf{x})$ is particularly relevant, then $v(t_1)$ might not be the best criterion to use. An alternative is to consider the Kullback-Leibler (KL) divergence between $\tilde{\pi}(\mathbf{x})$ and $q(\mathbf{x}|t_1)$, that is

$$\mathrm{KL}(t_1) = \int \log\left\{\frac{\tilde{\pi}(\mathbf{x})}{q(\mathbf{x}|t_1)}\right\}\tilde{\pi}(\mathbf{x}) d\mathbf{x} \quad (17)$$
$$\propto -\int \log\left\{q(\mathbf{x}|t_1)\right\}\frac{\pi(\mathbf{x})}{c} d\mathbf{x},$$

as done, in a related context, by Cappé et al (2008). The r.h.s. of (17), which we indicate with $g(t_1)$, can be estimated by

$$\hat{g}(t_1) = -\frac{1}{\hat{c}\, n_k}\sum_{i=1}^{n_k} \log\left\{q(\mathbf{x}_i|t_1)\right\} w_i, \quad (18)$$

where

$$\mathbf{x}_i \sim q(\mathbf{x}|t_1^I), \quad \text{for} \quad i = 1, \ldots, n_k,$$

with $\hat{c}$ and the w_is being defined as in (14).

To provide a simple illustration, we consider again the mixture target density of Section 5.2. In particular, we set $d = 5$ and we run LIMIS for $k = 50$ iterations, using a grid of initial values for t_1^I. We then estimate the optimal value of t_1 by minimizing $\hat{\tilde{v}}(t_1)$ with $h(\mathbf{x}) = 1$. To reduce the computational effort, we compute (16) using a sub-sample of size $n_k/10$, drawn multinomially from the n_k available samples. The left plot in Figure 4 shows, for each value of t_1^I, the estimated t_1^*, averaged over 60 runs. After estimating t_1^* for each t_1^I, we use each of the resulting mixture densities within an importance sampler, and we evaluate its efficiency. The average efficiencies of $q(\mathbf{x}|t_1^*)$ and $q(\mathbf{x}|t_1^I)$ are compared in the right plot of Figure 4. Optimizing over t_1 brings about drastic improvements in efficiency, if t_1^I is set too low. This is to be expected, because for low t_1 the importance mixture density is composed of widely spaced and narrow modes, which leads to highly variable weights. Notice that, if t_1^I is set too low, $q(\mathbf{x}|t_1^I)$ might not dominate $q(\mathbf{x}|t_1)$ or $\pi(\boldsymbol{x})$. If this is the case or, more generally, if the behaviour of $\hat{v}(t_1)$ or $\hat{\tilde{v}}(t_1)$ on the grid seems unstable, it might be appropriate to compute these quantities using Truncated Importance Sampling (Ionides, 2008).

8 Conclusions

The LIMIS algorithm provides a simple but flexible iterative framework for concurrently constructing a mixture importance density and performing importance sampling using such a density. By exploiting the shape information about the target density, LIMIS scales well with the dimensionality of the sampling space, especially if compared with the original NIMIS algorithm. The examples show that the performance of LIMIS compares favourably with that of a state-of-the-art MCMC

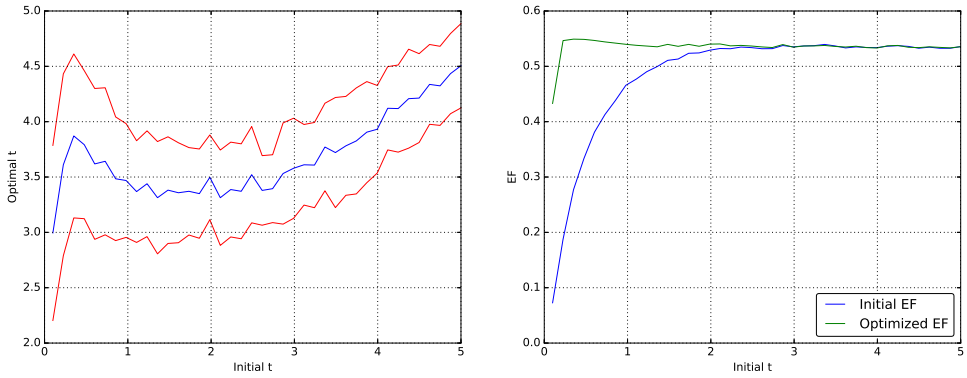

Fig. 4 Left: mean ($\pm\sigma$) optimal pseudo-time t_1^*, as a function of initialization t_1^I. Right: mean efficiency (EF) when the mixture derived using the initial, t_1^I, or the optimized, t_1^*, pseudo-time is used for importance sampling.

sampler such as MALA, under a multimodal, a nearly Gaussian and a ridge-like target.

Notice that, as the number of iterations increases, the covariance matrices produced by NIMIS become more localized, because the space is filled with more candidate neighbours. In contrast, the final pseudo-time t_1, which controls the location shrinkage and the covariance expansion of the mixture components produced by LIMIS, does not vary with the number of iterations. In Section 7 we showed how a single t_1 can be selected by minimizing a function-specific variance estimate. This requires post-processing the results of a preliminary LIMIS run. A promising direction for future research might be using a different value of t_1 for each new mixture component, and selecting it in an adaptive fashion. For highly non-Gaussian targets, we expect that the optimal t_1 would be quite large for the first mixture components, but it would then decrease as more localized components are added. Cappé et al (2008) select weights and parameters of a mixture of Gaussian or multivariate Student's t densities, by adaptively minimizing an entropy criterion. We think that their approach could be adjusted to fit our context. The resulting adaptive algorithm should benefit greatly from the fact that the locations and covariance matrices of LIMIS mixture components are entirely controlled by the target's shape and by t_1, which would drastically reduce the number of parameters that need to be optimized during the adaptation step.

Acknowledgements The authors would like to thank Samuel Livingstone and two anonymous referees for providing useful comments on an earlier version of this paper.

References

Ascher UM, Petzold LR (1998) Computer methods for ordinary differential equations and differential-algebraic equations. Society for Industrial and Applied Mathematics

Bates S (2001) Bayesian inference for deterministic simulation models for environmental assessment. PhD thesis, University of Washington

Bezanson J, Karpinski S, Shah VB, Edelman A (2012) Julia: A fast dynamic language for technical computing. arXiv:12095145

Brent RP (2013) Algorithms for minimization without derivatives. Courier Corporation, North Chelmsford

Bucy RS, Joseph PD (1987) Filtering for stochastic processes with applications to guidance. American Mathematical Society

Bunch P, Godsill S (2016) Approximations of the optimal importance density using Gaussian particle flow importance sampling. Journal of the American Statistical Association 111(514):748–762

Cappé O, Douc R, Guillin A, Marin JM, Robert CP (2008) Adaptive importance sampling in general mixture classes. Statistics and Computing 18(4):447–459

Carpenter B, Gelman A, Hoffman M, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017) Stan: a probabilistic programming language. Journal of Statistical Software 76(1)

Daum F, Huang J (2008) Particle flow for nonlinear filters with log-homotopy. In: SPIE Defense and Security Symposium, International Society for Optics and Photonics, pp 696,918–696,918

Duane S, Kennedy AD, Pendleton BJ, Roweth D (1987) Hybrid Monte Carlo. Physics Letters B 195(2):216–222

Faes C, Ormerod JT, Wand MP (2011) Variational Bayesian inference for parametric and nonparametric

regression with missing data. Journal of the American Statistical Association 106(495):959–971

Girolami M, Calderhead B (2011) Riemann manifold Langevin and Hamiltonian Monte Carlo methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73(2):123–214

Givens GH, Raftery AE (1996) Local adaptive importance sampling for multivariate densities with strong nonlinear relationships. Journal of the American Statistical Association 91(433):132–141

Gorman RP, Sejnowski TJ (1988) Analysis of hidden units in a layered network trained to classify sonar targets. Neural Networks 1(1):75–89

Haario H, Saksman E, Tamminen J (2001) An adaptive Metropolis algorithm. Bernoulli 7(2):223–242

Hoffman MD, Gelman A (2014) The No-U-Turn Sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. Journal of Machine Learning Research 15(1):1593–1623

Ionides EL (2008) Truncated importance sampling. Journal of Computational and Graphical Statistics 17(2):295–311

Kong A, Liu JS, Wong WH (1994) Sequential imputations and Bayesian missing data problems. Journal of the American Statistical Association 89(425):278–288

Lichman M (2013) UCI machine learning repository. URL http://archive.ics.uci.edu/ml

Raftery AE, Bao L (2010) Estimating and projecting trends in HIV/AIDS generalized epidemics using incremental mixture importance sampling. Biometrics 66(4):1162–1173

Roberts GO, Tweedie RL (1996) Exponential convergence of Langevin distributions and their discrete approximations. Bernoulli 2(4):341–363

Roberts GO, Rosenthal JS, et al (2001) Optimal scaling for various Metropolis-Hastings algorithms. Statistical Science 16(4):351–367

Schuster I (2015) Gradient importance sampling. arXiv:150705781

Sim A, Filippi S, Stumpf MP (2012) Information geometry and sequential Monte Carlo. arXiv:12120764

Süli E, Mayers DF (2003) An introduction to numerical analysis. Cambridge University Press, Cambridge

West M (1992) Modelling with mixtures. In: Berger J, Bernardo J, Dawid A, Smith A (eds) Bayesian Statistics, vol 4, Oxford University Press, Oxford, pp 503–525