INTEGRATION WITH AN ADAPTIVE HARMONIC MEAN ALGORITHM

ALLEN CALDWELL, RAFAEL C. SCHICK, OLIVER SCHULZ, MARCO SZALAY

MAX PLANCK INSTITUTE FOR PHYSICS, MUNICH, GERMANY

ABSTRACT. Efficiently estimating the integral of functions in high dimensional spaces is a non-trivial task when an analytical solution cannot be calculated. A oft-encountered example is in the calculation of the marginal likelihood, in a context where a sampling algorithm such as a Markov Chain Monte Carlo provides samples of the function. We present the Adaptive Harmonic Mean Integration (AHMI) algorithm. Given samples drawn according to a probability distribution proportional to the function, the algorithm will estimate the integral of the function and the uncertainty of the estimate by applying a harmonic mean estimator to adaptively chosen subvolumes of the parameter space. We describe the algorithm and it's mathematical properties, and report the results using it on multiple test cases of up to 20 dimensions.

1. INTRODUCTION

Sampling algorithms such as Markov Chain Monte Carlo (MCMC) algorithms [1] are often used to generate samples distributed according to non-trivial densities in high dimensional spaces. Many algorithms have been developed that allow generating samples Λ from an unnormalized target density $f(\lambda)$:

$$\Lambda \sim f(\lambda) \qquad f(\lambda) \geq 0 \quad .$$

In many applications, it is desirable or even necessary to be able to normalize the target density, i.e., to calculate

$$\tag{1} I \equiv \int_\Omega f(\lambda) d\lambda$$

where Ω is the support of f. This integral can be computationally very costly or impossible to perform with standard techniques; e.g., if the volume where the target f is non-negligible occupies a very small part of the total volume.

An important area where such integration is necessary is Bayesian inference [2, 3]. Bayes' formula, for a given model M, is

$$\tag{2} P(\lambda|\text{Data}, M) = \frac{P(\text{Data}|\lambda, M)P_0(\lambda|M)}{P(\text{Data}|M)}$$

where λ denotes the parameters of the model and the data are used to update probabilities for possible values of λ from prior probabilities $P_0(\lambda|M)$ to posterior probabilities $P(\lambda|\text{Data}, M)$. The denominator is usually expanded using the Law of Total Probability and written in the form

$$\tag{3} Z = P(\text{Data}|M) = \int P(\text{Data}|\lambda, M)P_0(\lambda|M)d\lambda.$$

Z goes by the name 'evidence', or 'marginal likelihood', and is an example of the type of integral that we want to be able to calculate (here the data are fixed and $f(\lambda) = P(\text{Data}|\lambda, M)P_0(\lambda|M)$). An example use of Z is the calculation of a Bayes factor to compare two models:

$$\text{BF} \equiv \frac{P(\text{Data}|M_A)}{P(\text{Data}|M_B)} = \frac{Z_A}{Z_B} \quad .$$

Another application where the calculation of a normalization can be very important is the parallelization of MCMC algorithms [4, 5]. While the MCMC approach is very attractive in many respects, it is often computationally expensive. It is therefore desirable to parallelize the task of mapping out the target density. This looks at first sight difficult since all MCMC algorithms are by construction serial. A parallelization of the calculations can however be achieved via a partitioning of the support. I.e., we partition Ω into subvolumes ω_i with

$$\cup \omega_i = \Omega \qquad\qquad \omega_i \cap \omega_j = \emptyset \text{ for } i \neq j$$

and we run a separate MCMC sampling for each subvolume ω_i. In order to have a final set of samples representing the target density over the full support, we need to know the relative probabilities for the different subvolumes. I.e, we need

$$I_i \equiv \int_{\omega_i} f(\lambda) d\lambda \qquad\qquad \left(\sum_i I_i = I \right)$$

The samples in the different regions are then given weights

$$w_i \propto \frac{I_i N}{I N_i}$$

with N_i the number of samples from $f(\lambda)$ in ω_i and $\sum_i N_i \equiv N$.

We are specifically interested in providing an algorithm applicable in a setting where samples are available from the target density $f(\lambda)$ but with possibly no further recourse to generating more samples. For this purpose, we investigate the use of a modified harmonic mean estimator (HME). We introduce the use of a reduced integration volume to improve the HME performance. After a description of the technique, we report on numerical investigations using Metropolis-Hastings MCMC samples. Our work has in common with [6] the use of the ratio of samples in a subvolume to the total number to scale up an integral in a subvolume, but we use a substantially different integral estimation technique.

2. Reduced Volume HME

We are interested in calculating I from Eq. 1. We start by defining

$$(4) \qquad\qquad r \equiv \frac{\int_\Delta f(\lambda) d\lambda}{I}$$

with $\Delta \subset \Omega$ a finite volume integration domain. Given our assumption that the sampling algorithm has successfully sampled from $f(\lambda)$, we can define the unbiased estimator

$$(5) \qquad\qquad \hat{r} \equiv \frac{N_\Delta}{N_\Omega}$$

which is available by counting the fraction of samples falling within $\Delta \subset \Omega$. I.e., the task of evaluating I reduces to integrating the function $f(\lambda)$ over a well-chosen region (or several regions) - presumably small regions around local modes, and multiplying the integral with $1/\hat{r}$.

Defining

$$(6) \qquad\qquad I_\Delta \equiv \int_\Delta f(\lambda) d\lambda$$

and

$$(7) \qquad\qquad \tilde{f}_\Delta(\lambda) = \frac{f(\lambda)}{I_\Delta} \qquad \lambda \in \Delta$$

we perform a harmonic mean calculation as follows:

$$(8) \qquad E\left[\frac{1}{f(\lambda)}\right]_{\hat{f}_\Delta(\lambda)} = \int_\Delta \frac{1}{f(\lambda)} \cdot \frac{f(\lambda)}{I_\Delta} d\lambda$$

$$(9) \qquad = \frac{V_\Delta}{I_\Delta}$$

where V_Δ is the volume of the space defined by Δ. The HME is then

$$(10) \qquad \hat{I}_\Delta \equiv \frac{N_\Delta V_\Delta}{\sum_{\Lambda_i \in \Delta} \frac{1}{f(\Lambda_i)}} \ .$$

This calculation is performed directly from the values of the target density $f(\Lambda_i)$ given by the sampling algorithm, and does not require any extra sampling. We rewrite the estimate as follows

$$(11) \qquad \hat{I}_\Delta = \frac{f_{\min} N_\Delta V_\Delta}{\sum_{\Lambda_i \in \Delta} \frac{f_{\min}}{f(\Lambda_i)}}$$

to highlight the importance of small function values which can dominate the sum in the denominator. In our approach, the value of $f_{\min}$ is controlled by the choice of the integration region Δ, which is chosen such that the variance of the denominator is bounded.

The estimator for the integral of interest is

$$(12) \qquad \hat{I} \equiv \frac{\hat{I}_\Delta}{\hat{r}} = \frac{f_{\min} N_\Omega V_\Delta}{\sum_{\Lambda_i \in \Delta} \frac{f_{\min}}{f(\Lambda_i)}} \ .$$

The initial volume over which the integration is performed can be large and even infinite and does not affect the final integral evaluation. We discuss the bias and uncertainty of this estimator below.

In general, samples come with weights (e.g., many algorithms such as MCMC repeat samples, and in this case the weight is the number of repetitions). We therefore generalize Eq. (12) to

$$(13) \qquad \hat{I} = \frac{f_{\min} W_\Omega V_\Delta}{\sum_{\Lambda_i \in \Delta} w_i \frac{f_{\min}}{f(\Lambda_i)}}$$

where w_i is the weight assigned to sample at parameter value Λ_i and $W_\Omega = \sum_i w_i$ is the sum of all weights.

2.1. **Relation to other techniques for evidence calculation.** A variety of techniques to calculate the evidence in Bayesian Calculations have been successfully developed. A summary can be found in [7], where a number of MCMC related techniques are reviewed, including Laplace's method [8], harmonic mean estimation [9], Chib's method [10], annealed importance sampling techniques [11, 12], nested sampling [13] and thermodynamic integration methods [14, 15]. Only the HME and Laplace techniques allow the direct estimation of the evidence from available samples, and the Laplace technique makes the unwanted assumption that the target density is a multi-variate Gaussian.

In the Bayesian literature [9], the HME for the evidence, Z, is formulated as

$$(14) \qquad Z = \frac{1}{E\left[\frac{1}{P(\text{Data}|\lambda,\text{M})}\right]_{P(\lambda|\text{Data},M)}}$$

$$(15) \qquad = \frac{1}{\int \frac{P(\lambda|\text{Data},M)}{P(\text{Data}|\lambda,\text{M})} d\lambda} \ .$$

The difference with our formulation is that the prior $P_0(\lambda|M)$ is not included in the expectation value, and the integration volume does not appear. In order to perform this HME calculation, it is

necessary to know separately the value of the likelihood and the prior at the sampling points. This method has been strongly criticized (even called 'worst Monte Carlo Method ever' [16]), since the evaluation of the denominator in Eq. (14) can have very large variance. Reducing the variance by limiting the integration volume is not as straightforward as in our formulation since it requires an integral of the prior function over the reduced volume.

2.2. **Bias and Uncertainty of the estimator.** We can estimate the bias and uncertainty on $\hat{I}$ given in Eq. (12) by separately analyzing the behavior of $\hat{r}$ and $\hat{X} = \frac{1}{N_\Delta} \sum_{\Lambda_i \in \Delta} \frac{1}{f(\Lambda_i)}$. As described, we choose regions Δ for which the range of target density values is moderate, so that the Central Limit Theorem applies to $\hat{X}$ and, for large enough N_Δ, we have that

$$P(\hat{X}) \approx \mathcal{N}\left(\mu = E\left[\frac{1}{f(\lambda)}\right], \sigma^2 = \frac{V\left[\frac{1}{f(\lambda)}\right]}{N_\Delta}\right)$$

where $P(\hat{X})$, the probability distribution for $\hat{X}$, is approximately given by a normal distribution with mean $\mu = E\left[\frac{1}{f(\lambda)}\right]$ and variance $\sigma^2 = \frac{V\left[\frac{1}{f(\lambda)}\right]}{N_\Delta}$. We can therefore estimate the uncertainty from $\hat{X}$ directly from the observed values of f. Given the asymmetry in the distribution of $\frac{1}{\hat{X}}$ we expect a bias of our integral estimate to larger values. However, this can be controlled by limiting the variance of $\hat{X}$ and is negligible in the examples we have considered.

In *iid* sampling, we would expect $\hat{r}$ to be unbiased and the uncertainty from $\hat{r}$ to be given by Binomial fluctuations. However, many samplers such as MCMC algorithms generate strong correlations amongst samples and using the binomial uncertainty is often inaccurate. It is therefore necessary to numerically evaluate the uncertainty. We do this by subdividing the set of samples into subsets, evaluate $\hat{I}$ for each subset and compute a standard error of the mean value of $\hat{I}$. Our procedure is described in detail below.

3. Adaptive Harmonic Mean Integration Algorithm

The Adaptive Harmonic Mean Integration (AHMI) algorithm uses the reduced volume HME on multiple subvolumes Δ_i to estimate the integral of $f(\lambda)$ over the full support. In this section we will present the algorithm in detail. As discussed previously, defining a set of suitable subvolumes is essential to obtain a robust and unbiased estimate of the integral of $f(\lambda)$. In particular, to avoid biasing the result, it is essential not to use the same elements of the MCMC set $\{\Lambda\}$ to both define the subvolumes Δ_i and estimate the integral $\hat{I}_i$.

3.1. **Finding Subvolumes.** The task of generating the subvolumes Δ_i is carried out by the AHMI algorithm as follows:

(1) The sample set $\{\Lambda\}$ is split into two separate subsets, $\{\Lambda^A\}$ and $\{\Lambda^B\}$.
(2) A whitening transformation, calculated based on Λ^A, is applied on Λ^A and Λ^B, resulting in a covariance matrix equal to the identity matrix.
(3) Samples with large $f(\lambda)$ are selected as seeds for the creation of the subvolumes Δ_i. Minimum-distance requirements between the seeds are enforced to avoid excessive clustering of the seeding samples and a consequent significant overlap of the generated subvolumes.
(4) A small hyper-cube is created around each seed.
(5) A user-selectable threshold factor f_p is defined; hyper-cubes for which the ratio of function values $\frac{f_{\max}}{f_{\min}} > f_p$ are rejected.
(6) For the M remaining hyper-cubes, each face is adjusted individually to adapt to the density of the samples. This deforms the cubes into hyper-rectangles. During all volume transformations, the condition $\frac{f_{\max}}{f_{\min}} \leq f_p$ is enforced.

(7) The hyper-rectangle-shaped subvolumes, created from samples $\in \Lambda^A$ (resp. Λ^B), are populated exclusively with independent samples obtained from Λ^B (resp. Λ^A).

3.2. Integral estimator.

Once the subvolumes are defined, we can compute the integral estimate $\hat{I}_i$ on each subvolume Δ_i, according to eq. 13.

To increase the stability of the algorithm, a cut is applied around the median of the distribution of all $\hat{I}_i$, removing the subvolumes associated with integral values outside of the cut. The final HME integral and an estimate of its uncertainty is computed using a robust and unbiased estimator for the combination of correlated measurements, as suggested in [17]:

$$(16) \qquad \hat{I} = \sum_{i=1}^{M} w_i \hat{I}_i$$

with

$$(17) \qquad w_i = \frac{\frac{1}{\bar{\sigma}_{i,i}^2}}{\sum_{j=1}^{M} \frac{1}{\bar{\sigma}_{j,j}^2}}$$

where $\bar{\sigma}_{i,i}^2$, the variance of the mean assigned to one subvolume, appear in the diagonal of the covariance matrix defined below. Note that the value of M here is smaller than that defined in the previous section due to the pruning of tail values of $\hat{I}_i$. In the following, M represents the number of subvolumes after pruning.

The estimated variance of $\hat{I}$ is

$$(18) \qquad \sigma^2(\hat{I}) = \sum_{i,j=1}^{M} w_i w_j \Sigma_{i,j}$$

where $\Sigma_{i,j}$ is the covariance matrix of $\hat{I}_i$ measurements over the M subvolumes Δ_i. Since the exact covariance matrix is unknown, in the following we introduce the required steps to obtain an estimate of $\Sigma_{i,j}$.

3.3. Estimated covariance matrix.

We use the following procedure to estimate $\Sigma_{i,j}$. The procedure is the same for $\{\Lambda^A\}$ and $\{\Lambda^B\}$, so we shall drop the superscript in this section.

(1) We begin by partitioning $\{\Lambda\}$ into subsets chosen in a way that reduces their correlation. A number S of subsets $\{\Lambda_1\}$, $\{\Lambda_2\}$ ... $\{\Lambda_S\}$ is defined from interleaved groups of samples. Each group has a total weight that is user-selectable and defaults to 100.

(2) An estimate of the integral $\hat{I}$ is then performed for all combinations of hyper-rectangles Δ_i and sample subset Λ_j resulting in a $S \times M$ matrix Y of correlated integral estimates:

$$Y = \begin{bmatrix} \hat{I}_{1,1} & \hat{I}_{1,2} & ... & \hat{I}_{1,M} \\ \hat{I}_{2,1} & \hat{I}_{2,2} & ... & \hat{I}_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{I}_{S,1} & \hat{I}_{S,2} & ... & \hat{I}_{S,M} \end{bmatrix}$$

(3) For each hyper-rectangle Δ_i, the variance σ_i^2 of the integrals $\hat{I}_{j,i}$ is calculated from Y using the distribution of integral estimation over the subset S:

$$\sigma_i^2 = \frac{1}{S-1} \sum_{j=1}^{S} \left(\hat{I}_{j,i} - \hat{I}_i \right)^2 .$$

with

$$\hat{I}_i = \frac{1}{S} \sum_{j=1}^{S} \hat{I}_{j,i}$$

(4) The estimate of the variance of $\hat{I}_i$ is then $\bar{\sigma}_i^2 = \frac{\sigma_i^2}{S}$.
(5) We found that the covariance of Y misrepresents the correlation between hyper-rectangles, probably because it underestimates the overlap of the hyper-rectangles and, thus, the off-diagonal elements of Σ. A better estimate of Σ is

$$\Sigma = \begin{bmatrix} \bar{\sigma}_{1,1}^2 & \rho_{1,2}\bar{\sigma}_1\bar{\sigma}_2 & \cdots & \rho_{1,M}\bar{\sigma}_1\bar{\sigma}_M \\ \rho_{2,1}\bar{\sigma}_2\bar{\sigma}_1 & \bar{\sigma}_2^2 & \cdots & \rho_{2,M}\bar{\sigma}_2\bar{\sigma}_M \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{M,1}\bar{\sigma}_M\bar{\sigma}_1 & \rho_{M,2}\bar{\sigma}_M\bar{\sigma}_2 & \cdots & \bar{\sigma}_M^2 \end{bmatrix}$$

The correlations $\rho_{i,j}$ are estimated from the number of weighted samples shared by two subvolumes Δ_i, Δ_j as follows:

$$\rho_{i,j} = \frac{W_\cap(i,j)}{W_\cup(i,j)}$$

with $W_\cap(i,j)$ and $W_\cup(i,j)$ the number of weighted samples in $\Delta_i \cap \Delta_j$ and $\Delta_i \cup \Delta_j$ respectively.

3.4. **Final estimate.** As we obtain two values of $\hat{I}$ and two variances $\sigma^2(\hat{I})$ from the two sets $\{\Lambda^A\}$ and $\{\Lambda^B\}$, we calculate the final result as

$$\hat{I} = \frac{\hat{I}_A/\sigma_A^2 + \hat{I}_B/\sigma_B^2}{1/\sigma_A^2 + 1/\sigma_B^2}$$

with variance estimate

$$\sigma^2 = \left(\frac{1}{\sigma_A^2} + \frac{1}{\sigma_B^2} \right)^{-1}.$$

A χ^2-test can be performed on the combination of the two estimates. Values greater than 1 can serve as a warning to diagnose poor convergence of the algorithm.

3.5. **Subvolume generation by example.** Now that the general flow of the algorithm has been outlined, we describe the subvolume generation steps in more detail, using samples from a two-dimensional Gaussian shell distribution as a visual aid, as shown in Fig. 1.
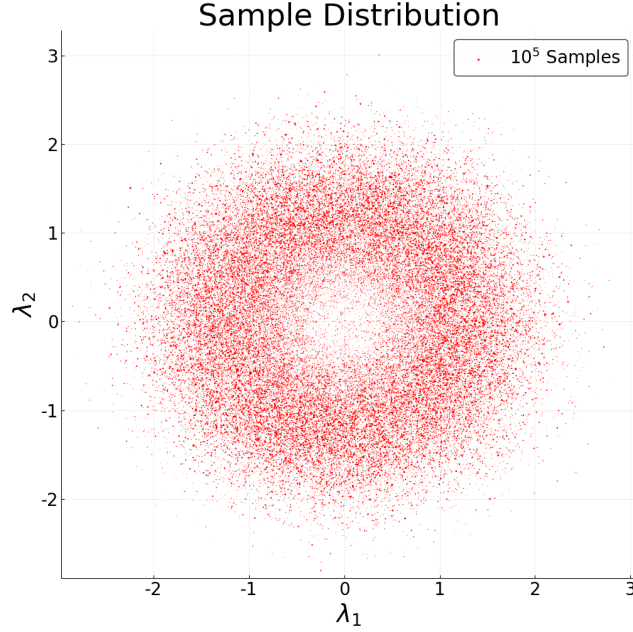
FIGURE 1. MCMC samples from a 2D-Gaussian shell distribution, after data whitening.

3.5.1. *Whitening Transformation.* In general, a whitening transformation maps a set of random variables with a known non-singular covariance matrix to a new set of variables with a covariance matrix equal to $\mathbb{I}$. A Cholesky Decomposition is used to whiten the samples, and the AHMI estimator for the integral becomes

$$(19) \qquad \hat{I} = \frac{f_{\min} W_\Omega V'_\Delta}{\det R \cdot \sum_{\Lambda'_i \in \Delta} w_i \frac{f_{\min}}{f(\Lambda'_i)}}$$

where $\det R$ is the determinant of the whitening matrix and the primed symbols represent the quantities in the transformed space.

3.5.2. *Creation of a Space-Partitioning Tree.* Efficiently identifying points (i.e. samples) in a point cloud that lie inside a particular hyper-rectangle is a nontrivial task. In order to reduce the computation time, a space-partitioning tree is used to divide the whitened space into subsets of non-overlapping regions with an equal number of samples inside them. An example of such space-partitioning tree is visualized in Fig. 2.
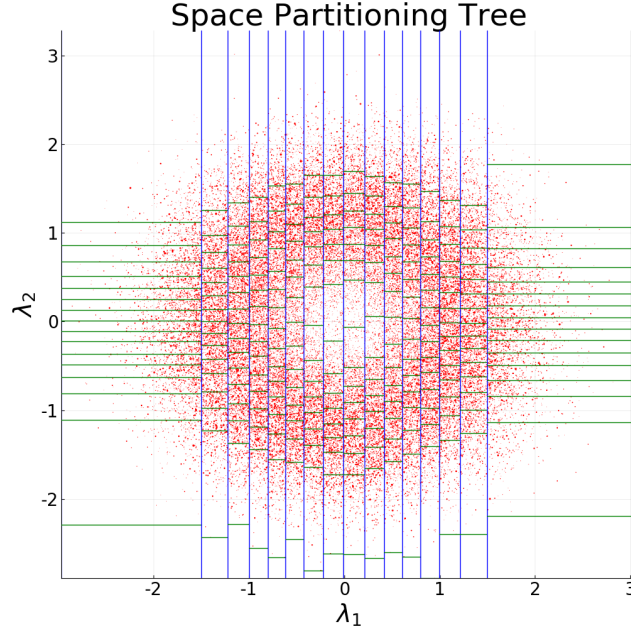
Space Partitioning Tree

FIGURE 2. Visualization of a 2D space-partitioning tree. All volumes contain an equal number of unique samples. The area of each point is proportional to the weight of that sample.

3.5.3. *Computation of the Probability Ratio Threshold.* In order to limit the variance and ensure numerical stability of the HME, the ratio between the highest probability and the lowest probability of samples inside a hyper-rectangle is bounded:

$$\text{(20)} \qquad \frac{f_{\max}}{f_{\min}} \le f_p \ .$$

Although this threshold is user-selectable, by default it is computed from the sample data and is defined by the ratio of the value of the function evaluated at the global mode, $f(\Lambda^*)$ and the value of the function at the 80th percentile, f_{80}.

$$\text{(21)} \qquad f_p = \frac{f(\Lambda^*)}{f_{80}} \ .$$

3.5.4. *Hyper-Rectangle Creation.* The following algorithm defines hyper-rectangle-shaped subvolumes, suitable for the AHMI, that adapt to the density of the samples and limit the variance of the HME.

Step 1: The algorithm starts by building a small hyper-cube around one of the previously selected seed samples, as shown in Fig. 3.
Step 2: Each hyper-cube is incrementally either increased or decreased, until its probability ratio matches the probability ratio threshold within a given tolerance or until it contains more than one percent of the total samples.
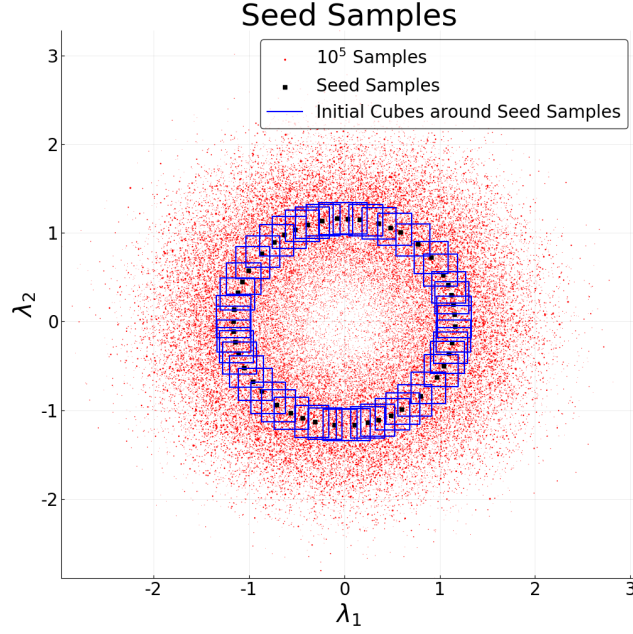
FIGURE 3. Display of the seed hypercubes.

Step 3: The faces of the hyper-cubes are iteratively adjusted, turning the D-dimensional hyper-cube into a D-dimensional hyper-rectangle, as follows:

```
while changes to the faces of the hyper-rectangle are made do
    for each dimension do
        while changes in this dimension are made do
            modify_lower_face()
            modify_upper_face()
        end
    end
end
```

The hyper-rectangle adaptation algorithm continues as long as changes to the hyper-rectangle's faces are accepted. The stopping criterion is based on the fraction of samples accepted or rejected compared to expectation from the volume change. However, the hyper-rectangle adaption algorithm always ensures that no modification to the hyper-rectangle's faces are made if such a modification would result in $\frac{f_{\max}}{f_{\min}} > f_p$. Figure 4 highlights the set of hyper-rectangles that fulfill the selection criterion.
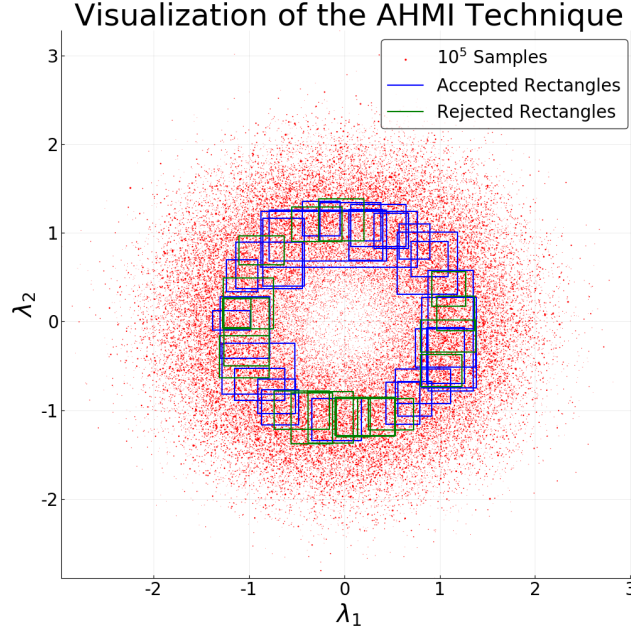
FIGURE 4. Hyper-rectangles after the seed hyper-cubes are adjusted. The blue hyper-rectangles are used in the final integral estimate.

4. EXAMPLES

To validate our algorithm, we performed the integration of several functions for which an accurate calculation of the integral is available. We start with a simple 1-dimensional Gaussian shell, and subsequently consider Gaussian shells of higher dimensionality as well as an example based on the multivariate normal distribution. The integration of each example is performed multiple times (K = 100) on separately generated MCMC samples. We compute the following quantities:

$$< \hat{I} > = \frac{1}{K} \sum_{k=1}^{K} \hat{I}_k$$

$$\text{stddev}(< \hat{I} >) = \frac{1}{\sqrt{K}} \sqrt{\sum_{k=1}^{K} \frac{\left(\hat{I}_k - < \hat{I} >\right)^2}{K-1}}$$

$$< \hat{\sigma} > = \frac{1}{K} \sum_{k=1}^{K} \sigma_k$$

$$\text{RMS}(\hat{\sigma}) = \sqrt{\sum_{k=1}^{K} \frac{(\sigma_k - < \hat{\sigma} >)^2}{K-1}}$$

with $\hat{I}_k$ and σ_k being the integration estimate and the corresponding standard deviation for the k-th occurrence of the calculation.

The AHMI values are compared with the known values to evaluate biases and the quality of our uncertainty estimation by considering the number of occurrences in which the estimated value lies within the estimated standard deviation from the true value (the expectation is that approximately 68 % of estimates will fall within one standard deviation).

4.1. **Gaussian shell.** Consider a function in D dimensions with degenerate modes lying on a $D-1$ dimensional surface of fixed radius, a Gaussian shell:

$$(22) \qquad f(\lambda | \vec{c}, r, \omega) = \frac{1}{\sqrt{2\pi\omega^2}} \exp\left(-\frac{(|\lambda - \vec{c}| - r)^2}{2\omega^2}\right) .$$

This function is centered at $\vec{c}$ with degenerate modes along a surface of radius r. The value of the function decreases when moving away from the modal surface along a radius, according to a Gaussian shape with standard deviation ω. The integral of this function can be easily evaluated numerically using spherical coordinates centered at $\vec{c}$, where ρ is the radial coordinate in the space, so that

$$I = \frac{1}{\sqrt{2\pi\omega^2}} \int \exp\left(-\frac{(\rho - r)^2}{2\omega^2}\right) dV$$

The volume element, integrated over the angular coordinates, is $dV = S_{D-1}\rho^{D-1}d\rho$ with $S_{D-1} = 2\pi^{D/2}/\Gamma(D/2)$, so that we have

$$I = \frac{\sqrt{2}\pi^{(D-1)/2}}{\Gamma(D/2)w} \int_0^{\rho_{max}} \rho^{D-1} \exp\left(-\frac{(\rho - r)^2}{2w^2}\right) d\rho .$$

We are left with a one-dimensional integral that can be calculated numerically to high precision. Note that we have assumed that the integral in the region outside $\rho_{\max}$ (the corners in the hypercube) is vanishingly small. This is the case for the examples considered in this work.

For the four examples below, we use the following settings: radius $r = 5$, width $\omega = 2$ and $\vec{c} = \vec{0}$. The integration region extends from $-25, +25$ in each dimension.

4.1.1. *1-Dimensional Gaussian shell.* For a 1-dimensional Gaussian shell, which is effectively the sum of two Gaussian distributions, the previously defined parameters result in $I_{\text{true}} \approx 1.98758$.

Our method produces the results summarized in table 1.

Samples	$< \hat{I} > \pm$ stddev	$< \hat{\sigma} >$	RMS($\hat{\sigma}$)	Results within σ_i	$< M >$
10^4	1.986 ± 0.003	0.021	0.006	65%	34
10^5	1.9876 ± 0.0008	0.0062	0.0020	65%	33
10^6	1.9879 ± 0.0003	0.0020	0.0006	58%	32

TABLE 1. Adaptive harmonic mean integration results for 100 estimations of a 1-dimensional Gaussian shell as a function of the number of MCMC samples. The integral value is $I_{\text{true}} = 1.9876$. The columns refer to the estimated integral value, the estimated standard deviation and its RMS, the fraction of occurrences where the true value was within 1σ of the estimate and the average number of hyper-rectangles generated to perform the integral estimation.

The AHMI method produces robust integral estimates and uncertainties, even with a relatively low number of MCMC samples. The fraction of occurrences in which the true value lies within the estimated standard deviation is not far from the expected value of 68.3%. For all iterations and number of MCMC samples, the χ^2 test introduced in section 3.4 is well below 1, suggesting that the method has converged properly.

4.1.2. *2-Dimensional Gaussian shell.* In two dimensions, the chosen parameters result in $I_{\text{true}} \approx$ 31.4411. The results produced by our method, averaged over 100 runs, are summarized in table 2.

Samples	$< \hat{I} > \pm$ stddev	$< \hat{\sigma} >$	RMS($\hat{\sigma}$)	Results within σ_i	$< M >$
10^4	31.23 ± 0.08	0.57	0.12	56%	49
10^5	31.43 ± 0.02	0.16	0.04	55%	99
10^6	31.438 ± 0.007	0.051	0.011	56%	155

TABLE 2. Adaptive harmonic mean integration results for 100 estimations of a 2-dimensional Gaussian shell as a function of the number of MCMC samples. The integral value is $I_{\text{true}} = 31.441$. The columns refer to the estimated integral value, the estimated standard deviation and its RMS, the fraction of occurrences where the true value was within 1σ of the estimate and the average number of hyper-rectangles generated to perform the integral estimation.

Again, the results are compatible with the true value within the estimated standard deviation and the χ^2 test shows good convergence of the method. To investigate the decrease in the fraction of iteration for which the true value lies with one standard deviation of the estimate (column 5 in table 2), we estimate the uncertainty on the integration again, using all the variances of all the hyper-rectangles of a single run to assess whether the small underestimation of σ is due to the underestimation of the coverage of each individual hyper-rectangle or, rather, due to an underestimation of the correlation among hyper-rectangles. The results in table 3 hint at the latter.

Samples	$< \hat{I}_j > \pm$ stddev(I_j)	$< \hat{\sigma}_j >$	RMS($\hat{\sigma}_j$)	Results within σ_j
10^4	30.98 ± 0.26	3.04	1.56	75%
10^5	31.46 ± 0.12	1.30	0.53	71%
10^6	31.30 ± 0.03	0.25	0.07	62%

TABLE 3. Distribution of integral estimates I_j for all the hyper-rectangles of a single integration run over a 2-dimensional Gaussian shell.

4.1.3. *10-Dimensional Gaussian shell.* For the ten-dimensional Gaussian shell the actual value of the integral is $I_{\text{true}} \approx 1.1065 \cdot 10^9$. Again, we present our results in table 4 for different number of samples averaged over 100 runs. No bias is observed in the integral estimate, the uncertainty estimate has slightly smaller coverage than desired, again probably due to a slight underestimation of the correlation among the results from different hyper-rectangles.

Samples	$< \hat{I} > \pm$ stddev	$< \hat{\sigma} >$	RMS($\hat{\sigma}$)	Results within σ_i	$< M >$
10^4	$(1.060 \pm 0.023) \cdot 10^9$	$0.061 \cdot 10^9$	$0.021 \cdot 10^9$	53%	54
10^5	$(1.105 \pm 0.002) \cdot 10^9$	$0.012 \cdot 10^9$	$0.003 \cdot 10^9$	52%	115
10^6	$(1.1064 \pm 0.0006) \cdot 10^9$	$0.0040 \cdot 10^9$	$0.0016 \cdot 10^9$	50%	282

TABLE 4. Adaptive harmonic mean integration results for the 10-dimensional Gaussian shell. The integral value is $I_{\text{true}} = 1.1065 \cdot 10^9$. The columns refer to the estimated integral value, the estimated standard deviation and its RMS, the fraction of occurrences where the true value was within 1σ of the estimate and the average number of hyper-rectangles generated to perform the integral estimation.

4.1.4. *20-Dimensional Gaussian shell.* We present the 20 dimensional case as an extreme test case scenario for our integration technique. The expected result is: $I_{\text{true}} \approx 2.9772 \cdot 10^{17}$. Table 5 summarizes the results from our method.

Samples	$< \hat{I} > \pm$ stddev	$< \hat{\sigma} >$	RMS($\hat{\sigma}$)	Results within σ_i	$< M >$
10^5	$(3.245 \pm 0.016) \cdot 10^{17}$	$0.094 \cdot 10^{17}$	$0.020 \cdot 10^{17}$	14%	153
10^6	$(3.062 \pm 0.010) \cdot 10^{17}$	$0.028 \cdot 10^{17}$	$0.007 \cdot 10^{17}$	25%	298

TABLE 5. Adaptive harmonic mean integration results for the 20-dimensional Gaussian shell for 100 integration runs. The integral value is $I_{\text{true}} = 2.977 \cdot 10^{17}$. The columns refer to the estimated integral value, the estimated standard deviation and its RMS, the fraction of occurrences where the true value was within 1σ of the estimate and the average number of hyper-rectangles generated to perform the integral estimation.

In this case the method produces biased results when the number of MCMC samples is not high enough. This is due to not having enough samples in the corners of the hyper-rectangles that are used to estimate the integral: areas that are poorly populated are biased towards higher values of $\hat{I}$, as shown in eq. 13. In addition, since the uncertainty estimation is calculated by splitting the samples into subsets, it is not surprising to see it fail when the sample set is limited. Numerically integrating the Gaussian shell in higher dimensions is a particularly demanding test of our algorithm since the shape of the volumes is a poor match to the density of the samples. We plan to allow the user to select other volume shapes, e.g. spheroids or shells, to better approximate their particular problem in the future. We would also like to note that the aforementioned results are already comparable to or better than other widely used integration libraries. As an example, in this scenario, all methods in the CUBA integration library [18] fail to converge.

4.2. **Multivariate Normal Distribution.** The target function in this case is:

$$(23) \qquad f(\vec{x}) = \frac{1}{(2\pi)^5 |\Sigma|^{1/2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1} (\vec{x}-\vec{\mu})}$$

where Σ is the covariance matrix, given below. The target function is 10-dimensional and has significant correlations among the ten parameters. The mean vector $\vec{\mu}$ is $\vec{0}$.

$$\Sigma = \begin{bmatrix} 1.0 & 1.0 & 0.30 & 0.43 & -0.14 & -0.86 & -0.22 & -0.84 & 0.83 & -2.5 \\ 1.0 & 2.0 & -0.14 & 0.36 & 0.14 & -0.08 & -0.45 & 0.71 & 0.07 & -1.9 \\ 0.30 & -0.14 & 3.0 & -0.04 & 0.76 & 0.85 & 1.7 & -0.41 & 1.2 & -0.78 \\ 0.43 & 0.36 & -0.04 & 4.0 & -0.81 & -0.86 & -1.5 & -2.1 & 0.17 & 0.63 \\ -0.14 & 0.14 & 0.76 & -0.81 & 5.0 & 2.5 & 1.6 & 1.5 & 1.9 & 0.46 \\ -0.86 & -0.08 & 0.85 & -0.86 & 2.5 & 6.0 & 1.9 & 4.1 & -0.28 & 2.7 \\ -0.22 & -0.45 & 1.7 & -1.5 & 1.6 & 1.9 & 7.0 & 0.70 & 1.4 & 2.6 \\ -0.84 & 0.71 & -0.41 & -2.1 & 1.5 & 4.1 & 0.70 & 8.0 & -0.87 & 2.5 \\ 0.83 & 0.07 & 1.2 & 0.17 & 1.9 & -0.28 & 1.4 & -0.87 & 9.0 & -4.0 \\ -2.5 & -1.9 & -0.78 & 0.63 & 0.46 & 2.7 & 2.6 & 2.5 & -4.0 & 10 \end{bmatrix}$$

The analytical integral is equal to 1. The following table summarizes the results of AHMI as a function of the number of MCMC samples:

Samples	$< \hat{I} > \pm$ stderr	$< \hat{\sigma} > \pm$ stddev	Results within σ_i	$< M >$
10^4	0.978 ± 0.007	0.036 ± 0.010	49%	59
10^5	1.004 ± 0.004	0.012 ± 0.003	46%	97
10^6	1.002 ± 0.004	0.005 ± 0.001	42%	140

TABLE 6. Integration results for a multivariate normal distribution, averaged across 100 runs. The columns refer to the estimated integral value, the estimated standard deviation and its RMS, the fraction of occurrences where the true value was within 1σ of the estimate and the average number of hyper-rectangles generated to perform the integral estimation.

As we can see in this benchmark as well, our algorithm delivers robust results over a large span of number of available MCMC samples.

5. CONCLUSION

We have developed the Adaptive Harmonic Mean Integration (AHMI) algorithm that can be used to integrate a non-normalized density function using only the samples drawn according to the probability distribution proportional to the function. Assuming that the sampling algorithm has successfully produced samples from this distribution, the AHMI algorithm can be used to produce both an integral estimate as well as an estimate of the uncertainty of the integral. The algorithm has been tested on a number of examples and found to perform well. Negligible bias has been found in the examples considered, though the uncertainty is sometimes slightly underestimated. An exception was a high-dimensional distribution chosen to be ill-suited for the algorithm, here the quality of integral estimate is reduced when the sample size is limited. Diagnostics for the algorithm, based on the number of subvolumes generated, number of samples per subvolume and a χ^2 test of the statistically independent results are investigated in an attempt to identify biased or otherwise inconsistent results caused by poor MCMC statistics.

In the future, we aim to present the user with a warning when these situations arise. We also aim to investigate whether the estimate of the integral uncertainty can by improved by combining an estimate of the effective MCMC sample size with a semi-analytical prediction of the uncertainty (Eq. 13).

We believe the AHMI algorithm is ready to be used in many real-world applications and encourage our colleagues to provide feedback.

References

[1] See e.g., C. Robert and G. Casella, 'Monte Carlo Statistical Methods', 2nd Edition, Springer (2004).

[2] H. Jeffreys, 'Theory of Probability', 3rd Ed., Claredon Press, Oxford, MR0187257 (1961).

[3] E. T. Jaynes, 'Probability Theory: the logic of science', Cambridge University Press, Cambridge, MR1992316 (2003).

[4] D. N. VanDerwerken, S. C. Schmidler, 'Parallel Markov Chain Monte Carlo', arXiv:1312.7479v1

[5] A. Caldwell, C. Liu, 'Target Density Normalization for a Markov Chain Monte Carlo Algorithm', arXiv:1410.7149v2.

[6] M. D. Weinberg, 'Computing the bayes factor from a markov chain monte carlo simulation of the posterior distribution', *Bayesian Analysis*, 7 (2012) 737.

[7] N. Friel and J. Wyse, 'Estimating the evidence - a review', Stat. Neerl., **66** (2012) 2800.

[8] L. Tierney and J. B. Kadane, 'Accurate approximations for posterior moments and marginal densities', Journal of the American Statistical Associations, **81** (1986) 82.

[9] M. A. Newton and A. E. Raftery, 'Approximate Bayesian Inference with the Weighted Likelihood Bootstrap', Journal of the Royal Statistical Society, series B**56** (1994) 3.

[10] S. Chib, I. Jeliazkov, 'Marginal Likelihood from the Metropolis-Hastings Output', Journal of the American Statistical Association **96** (2001) 270.

[11] R. Neal, Journal of the Royal Statistical Society, Series B**56 (1)**, (1994) 41.

[12] C. P. Robert and D. Wraith, 'Computational Methods for Bayesian model choice', Bayesian Inference and Maximum Entropy Methods in Science and Engineering: the 29th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (AIP Conference Proceedings), vol. 1193 (2009) 251.

[13] J. Skilling, 'Nested Sampling for General Bayesian Computation', Bayesian Analysis **1** (2006) 833.

[14] A. Gelman and X. L. Meng, 'Simulating normalizing constants: from importance sampling to bridge sampling to path sampling', Statistical Science **13** (1998) 163.

[15] N. Friel and A. N. Pettitt, 'Marginal likelihood estimation via power posteriors', Journal of the Royal Statistical Society, Series B**70** (2008) 589.

[16] https://radfordneal.wordpress.com/2008/08/17/the-harmonic-mean-of-the-likelihood-worst-monte-carlo-method-ever/

[17] M. Schmelling, 'Averaging correlated data.' Physica Scripta 51.6 (1995): 676.

[18] T. Hahn. 'Cuba - a library for multidimensional numerical integration.' Computer physics communications 176, no. 11-12 (2007): 712-713.