International Journal of Modern Physics A
Vol. 35, No. 24 (2020) 2050142 (19 pages)
© The Author(s)
DOI: 10.1142/S0217751X20501420

World Scientific
www.worldscientific.com

Integration with an adaptive harmonic mean algorithm

Allen Caldwell,[*,‡] Philipp Eller,[†] Vasyl Hafych,[*] Rafael Schick,[*,†]
Oliver Schulz[*] and Marco Szalay[*,§]

[*] *Max Planck Institute for Physics, D-80805 Munich, Germany*
[†] *Technical University of Munich, D-80333 Munich, Germany*
[‡] *caldwell@mpp.mpg.de*

Received 24 March 2020
Revised 26 June 2020
Accepted 28 July 2020
Published 31 August 2020

Numerically estimating the integral of functions in high dimensional spaces is a nontrivial task. A oft-encountered example is the calculation of the marginal likelihood in Bayesian inference, in a context where a sampling algorithm such as a Markov Chain Monte Carlo provides samples of the function. We present an 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 regions of the parameter space. We describe the algorithm and its mathematical properties, and report the results using it on multiple test cases.

Keywords: Probability and statistics; integral estimation; evidence; MCMC.

PACS numbers: 02.70.Rr, 02.70.Uu

1. Introduction

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

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

[‡] Corresponding author.

[§] Now at Google LLC., Santa Barbara, CA, 93101, USA.

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

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

where Ω is the support of f. This integral can be computationally very costly or impossible to compute using standard techniques; e.g. if the volume of the support where the target f is non-negligible occupies a very small region of the full support.

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

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

where here 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

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

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:

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

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).[4] We introduce the use of a reduced integration region 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[5] the use of the ratio of samples found in the limited support region to the number found in the full support, but we use a substantially different integral estimation technique.

2. Reduced Volume HME

We are interested in estimating the integral I from Eq. (1). We start by defining the integral

$$I_\Delta \equiv \int_\Delta f(\lambda) d\lambda \tag{4}$$

with $\Delta \subset \Omega$ a finite integration region, and the ratio

$$r \equiv \frac{I_\Delta}{I}\,. \tag{5}$$

Given our assumption that the sampling algorithm has successfully sampled from $f(\lambda)$, we use the following as an estimator to our ratio

$$\hat{r} = \frac{N_\Delta}{N_\Omega}\,, \tag{6}$$

which is the fraction of the total number of samples that fall within $\Delta \subset \Omega$. Defining the normalized density over Δ

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

allows us to perform a harmonic mean calculation as follows:

$$E\left[\frac{1}{f(\lambda)}\right]_{\tilde{f}_\Delta(\lambda)} = \int_\Delta \frac{1}{f(\lambda)} \cdot \tilde{f}_\Delta(\lambda) d\lambda = \int_\Delta \frac{1}{f(\lambda)} \cdot \frac{f(\lambda)}{I_\Delta} d\lambda = \frac{V_\Delta}{I_\Delta}\,, \tag{8}$$

where V_Δ is the volume of the region defined by Δ. An estimator for this expectation value is the harmonic mean

$$\hat{X} = \frac{1}{N_\Delta} \sum_{\lambda_i \in \Delta} \frac{1}{f(\lambda_i)}\,. \tag{9}$$

The HME for the reduced volume integral then follows as

$$\hat{I}_\Delta = \frac{V_\Delta}{\hat{X}} = \frac{N_\Delta V_\Delta}{\sum_{\lambda_i \in \Delta} \frac{1}{f(\lambda_i)}}\,. \tag{10}$$

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. An estimator for the integral over the full space Ω can then be written down as

$$\hat{I} = \frac{\hat{I}_\Delta}{\hat{r}} = \frac{N_\Omega V_\Delta}{\sum_{\lambda_i \in \Delta} \frac{1}{f(\lambda_i)}}\,. \tag{11}$$

The task of estimating our integral, therefore reduces to choosing one or several subspaces Δ — typically small regions around local modes of $f(\lambda)$. The full space Ω over which the integration ought to be performed can be large or even infinite, while this does not affect the outcome of our integral estimate. We discuss the bias and uncertainty of this estimator in the following subsection.

In general, MCMC samples come with weights (e.g. repeated samples, with the weight being the number of repetitions). We therefore rewrite Eq. (11) as

$$\hat{I} = \frac{W_\Omega V_\Delta}{\sum_{\lambda_i \in \Delta} \frac{w_i}{f(\lambda_i)}} \tag{12}$$

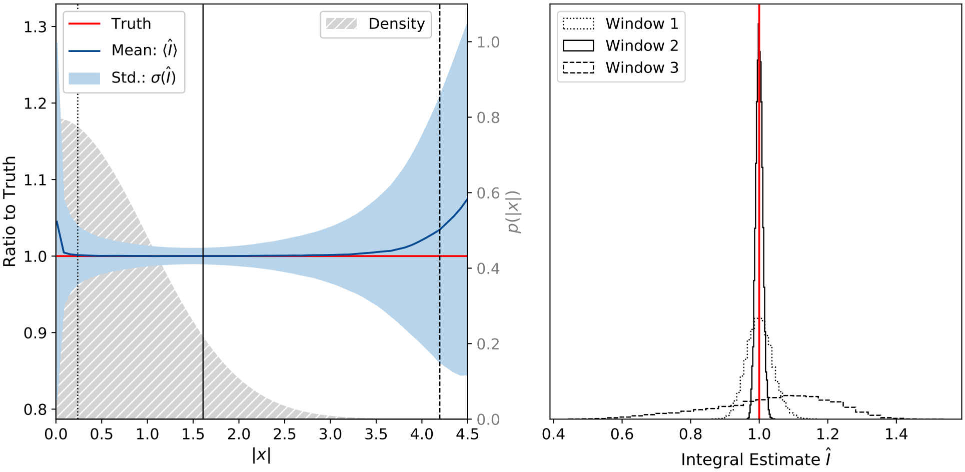


Fig. 1. (Color online) Demonstration of the reduced harmonic mean technique with the unit normal distribution. The left panel shows in blue the mean (solid line) and the region covering 68% of results (from $1.5 \cdot 10^4$ repeated trials) of the integral estimates as a function of the window extent. The true integral value is 1.0, indicated in red. The gray shaded distribution shows the unit normal pdf (right y-axis scale). For the three window extents indicated by the black, vertical lines (Window 1: $|x| < 0.24$, Window 2: $|x| < 1.61$, Window 3: $|x| < 4.20$) the full distribution of integral estimates from the $1.5 \cdot 10^4$ trials are plotted on the right side as histograms.

with w_i the weights assigned to the samples at parameter values λ_i and $W_\Omega = \sum_i w_i$ the sum of all weights. The use of weights also allows this technique to be applied to samples obtained from, for example, importance sampling.

2.1. *Illustration of the technique*

To illustrate our technique for applying harmonic mean integration, we consider the unit normal distribution $p(x) = \mathcal{N}(x|\mu = 0, \sigma = 1)$. A fixed number of samples ($3 \cdot 10^3$) was generated from directly sampling the unit normal distribution, and Eq. (11) was used to calculate the integral for different subregions Δ. These regions are defined as windows of x centered on 0 and varied in width from 0.02 up to 9. This was repeated $1.5 \cdot 10^4$ times, and the mean and standard deviation were evaluated. Figure 1 shows the results of the integration as a function of window size. As is seen in the figure, harmonic mean integration applied to a finite region gives an accurate value for the integral over a wide range of sampling windows. The variation in the integral results is largest for small windows due to the small number of samples used, and for large windows due to the divergence of the harmonic mean estimator. We discuss the biases in the next section.

2.2. *Bias and uncertainty of the estimator*

We can estimate the bias and uncertainty on $\hat{I}$ given in Eq. (11) by separately analyzing the behavior of $\hat{r}$ and $\hat{X}$. As described below, we choose regions Δ for which the range of target density values is moderate. For *i.i.d.* sampling, this would imply, via the Central Limit Theorem, that $\hat{X}$ follows a normal distribution. Assuming we can approximate the distribution of $\hat{X}$ with a normal distribution, we have

$$P(\hat{X}) \approx \mathcal{N}\left(\hat{\mu}_X = \hat{X}, \hat{\sigma}_X^2 = \frac{\sum_{\lambda_i \in \Delta}\left(\frac{1}{\lambda_i} - \hat{X}\right)^2}{N_\Delta(N_\Delta - 1)}\right),$$

where $P(\hat{X})$ is the probability distribution for $\hat{X}$ with mean μ and variance σ^2 estimated from the observed samples. Since $\hat{X}$ appears in the denominator in Eq. (11), this produces a bias in our integral of size $\hat{\sigma}_X^2/\hat{\mu}_X^2$. The fractional uncertainty in our integral estimator is $\hat{\sigma}_X/\hat{\mu}_X$.

The estimator $\hat{r}$ will also typically follow approximately a normal distribution with parameters that can be estimated from *i.i.d.* sampling and binomial statistics as

$$P(\hat{r}) \approx \mathcal{N}\left(\hat{\mu}_r = \hat{r}, \hat{\sigma}_r^2 = \frac{\hat{r}(1-\hat{r})}{N_\Omega}\right).$$

Since $\hat{r}$ also appears in the denominator, it will also produce a bias in our integral of size $\hat{\sigma}_r^2/\hat{\mu}_r^2$. The fractional uncertainty in our integral estimator from $\hat{r}$ is, in the approximation of *i.i.d.* sampling and a normal distribution, $\hat{\sigma}_r/\hat{\mu}_r$.

We can therefore write down an explicit correction factor

$$b = \left(1 - \frac{\hat{\sigma}_X^2}{\hat{\mu}_X^2} - \frac{\hat{\sigma}_r^2}{\hat{\mu}_r^2}\right) \tag{13}$$

that we apply to the integral estimate $\hat{I}$ by multiplication.

The correction is illustrated in Fig. 2 using the same numerical experiments discussed in Subsec. 2.1. The resulting uncorrected and corrected average integral values are displayed. Focusing on small window sizes, it can be seen that the term

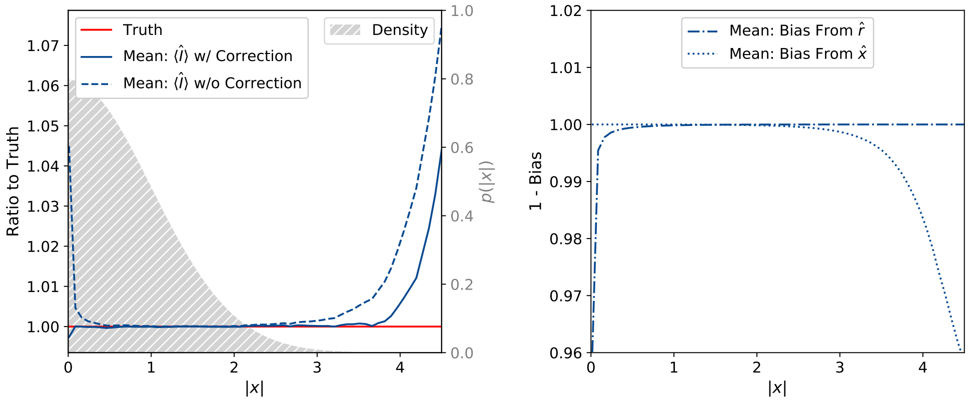


Fig. 2. Left: The average uncorrected (Eq. (11)) and corrected $(b \cdot \hat{I})$ integrals as a function of the extent of the window used to accept samples. The results are from averaging $1.5 \cdot 10^4$ integration results, where each integration test was performed from $3 \cdot 10^3$ samples in the full function range. Right: Individual contributions to the bias correction from the binomial $(\hat{r})$ and the $1/f$ terms.

from $\hat{r}$ dominates, as the binomial uncertainty is largest for small numbers of samples. With the bias correction applied, already for as few as ≈ 20 samples inside the integration volume (corresponding to roughly $|x| < 0.01$), accurate results are produced, while without the correction factor applied much large windows, starting at around $|x| < 0.5$, are necessary.

Toward larger windows the bias produced from $\hat{X}$ become dominant, as the range of values of f of the contained samples grows. The bias correction successfully mitigates this effect as illustrated.

Once the window size exceeds the space where samples are present, the integral starts diverging. For our example using $3 \cdot 10^3$ samples, we expect those to cover a region up to only $|x| \approx 3.58$, which explains well the observed trend.

Many samplers such as MCMC algorithms generate strong correlations among samples and using the binomial uncertainty discussed here can be inaccurate. We therefore also numerically evaluate the uncertainty as described in detail below. The integration regions are chosen such that the bias correction can be neglected.

2.3. *Relation to other techniques for evidence calculation*

A variety of techniques to calculate the marginal likelihood in Bayesian calculations have been successfully developed. A summary can be found in Ref. 8, where a number of MCMC related techniques are reviewed, including Laplace's method,[9] harmonic mean estimation,[4] 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 single multivariate Gaussian. The Laplace approximation fails badly for multimodal distributions or distributions with significant probability mass in the tails of the distribution.

In the Bayesian literature,[4] the HME for the evidence, Z, is formulated as follows:

$$\frac{1}{Z} = \int \frac{P(\lambda|\text{Data}, M)}{P(\text{Data}|\lambda, M)} d\lambda \tag{14}$$

$$= E\left[\frac{1}{P(\text{Data}|\lambda, M)}\right]_{P(\lambda|\text{Data},M)} . \tag{15}$$

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. The HME 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 region is not as straightforward as in our formulation since it requires an integral of the prior function over the reduced support whereas this is not needed in our formulation.

3. An Adaptive Harmonic Mean Integration Algorithm

Adaptive Harmonic Mean Integration (AHMI) uses the HME on multiple subregions Δ_i to estimate the integral of $f(\lambda)$ over its full support Ω. In this section we present our example algorithm in detail and will show benchmark tests on several distributions in Sec. 4. As discussed previously, defining a set of suitable regions is crucial in obtaining 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 sample set $\{\Lambda\}$ for both the definition of Δ_i and estimates of the integral $\hat{I}_i$.

(1) Start with samples $\{\Lambda\}$ from density $f(\lambda)$ and apply whitening transformation.

(2) Split into two mutually exclusive, uncorrelated and equally sized subsets A and B.

(3) Generate seed points λ_i^{seed} (via space partitioning tree).

(4) Create a small hyper-cube $\tilde{\Delta}_i$ around each seed point λ_i^{seed}.

(5) Adjust the faces of the hyper-cubes, resulting in hyper-rectangles.

(6) Use the resulting hyper-rectangles Δ_i^B created with sample $\{\Lambda^B\}$ for the harmonic mean estimates $\hat{I}_i^A$ on $\{\Lambda^A\}$ and vice versa.

(7) Use individual estimates $\hat{I}_i$ to compute combined estimate and variance for A and B.

(8) Compute final estimate by combining the independent estimates $\hat{I}_A$ and $\hat{I}_B$ weighted by their variance.

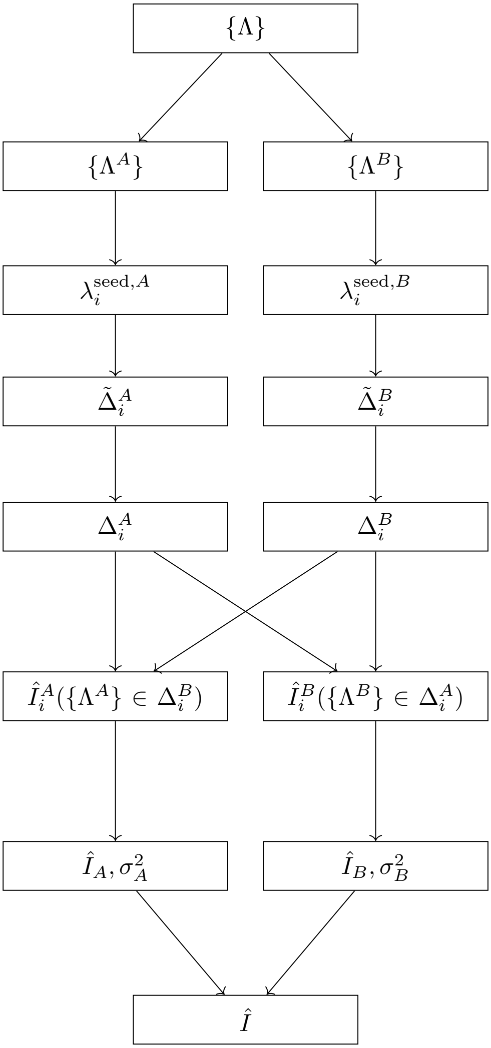


Fig. 3. Overview of the different steps in the AHMI algorithm, including the procedure of finding subvolumes Δ_i and computing integral estimates $\hat{I}_i$.

The general flow of the AHMI algorithm, including the procedure of defining the Δ_i, is summarized in Fig. 3. The various involved steps are discussed in more technical detail in the following subsections.

3.1. *Samples, preprocessing and splitting*

We start with a given set of samples $\{\Lambda\}$ that we assume are drawn according to the probability distribution proportional to our function f, obtained for example from MCMC sampling.

In order to de-correlate the sample space we apply a whitening transformation. In general, a whitening transformation maps a set of random variables with a known nonsingular 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

$$\hat{I} = \frac{W_\Omega V'_\Delta}{\det R \cdot \sum_{\lambda'_i \in \Delta} \frac{w_i}{f(\lambda'_i)}}\,, \tag{16}$$

where $\det R$ is the determinant of the whitening matrix and the primed symbols represent the quantities in the transformed space. In the following we drop this explicit addition of prime symbols and work in the whitened space (unless otherwise stated).

The full set of samples is then divided into two equally sized and mutually exclusive subsets A and B.

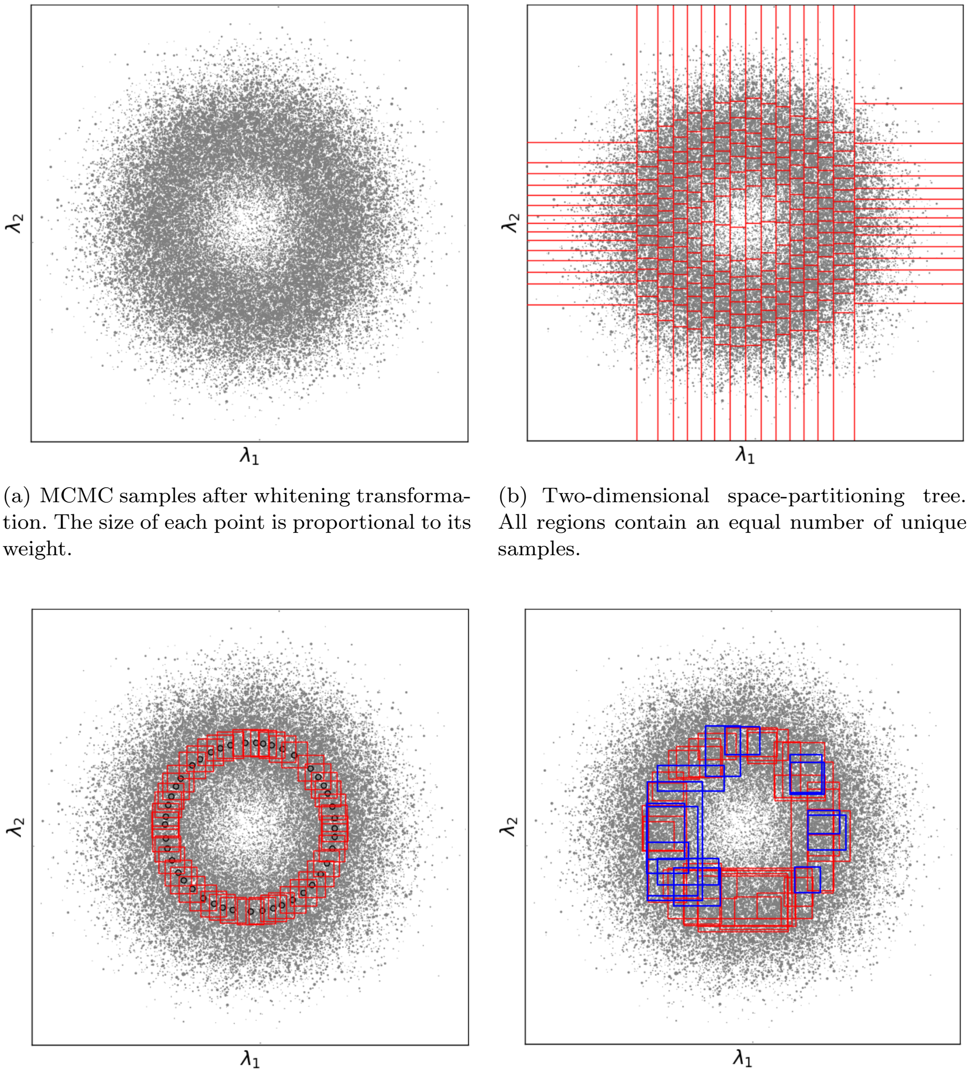
3.2. *Hyper-rectangle generation*

We illustrate the hyper-rectangle generation steps in more detail using a two-dimensional Gaussian shell example with distribution

$$f(\lambda|\vec{c}, r, \omega) = \frac{1}{\sqrt{2\pi\omega^2}} \exp\left(-\frac{(|\lambda - \vec{c}| - r)^2}{2\omega^2}\right). \tag{17}$$

In our examples, 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. Samples from this distribution are shown in Fig. 4(a).

The algorithm starts by creating seed points around which to construct the integration regions Δ_i. These points should lie in areas of high density and should result in broadly distributed starting points. In order to limit computation time, a simple space-partitioning tree is used to divide the whitened space into subsets of nonoverlapping regions. A tree is constructed by performing cuts in every dimension in such a way to have an equal number of samples on the left and right leaves. The number of cuts in each parameter axes is determined by the total number of samples and the number of dimensions and is chosen in a way, that the number of samples in each leaf does not exceed 200. An example of such a space-partitioning tree is



(a) MCMC samples after whitening transformation. The size of each point is proportional to its weight.

(b) Two-dimensional space-partitioning tree. All regions contain an equal number of unique samples.

(c) Initial hypercubes $\tilde{\Delta}$ around seed points.

(d) Hyper-rectangles after adjustments. Only the red hyper-rectangles are used in the final integral estimate.

Fig. 4. Process of finding integration regions in a two-dimensional example for the Gaussian shells test function.

shown in Fig. 4(b). For each partition i, the sample with the largest function value contained in that partition is defined as seed point $\lambda_i^{\mathrm{seed}}$.

The following steps produce hyper-rectangle-shaped regions suitable for AHMI. In order to limit this variance and ensure numerical stability, the ratio between the highest and the lowest probability of samples inside a hyper-rectangle is bound by the following condition:

$$\frac{f_{\max}}{f_{\min}} \leq t\,. \tag{18}$$

Although this threshold is user-selectable, by default it is equal to 500 (an example of integration with different threshold values is shown in Subsec. 4.1).

To create M regions for integration, we select the seed point λ_i^{seed} with the overall largest value $f(\lambda_i^{\text{seed}})$ and follow the steps below. Then we recursively repeat the same procedure $M-1$ times using the remaining seed points.

(1) The algorithm starts by building a small hyper-cube $\tilde{\Delta}_i$ around the selected seed point λ_i^{seed}, see Fig. 4(c).

(2) This hyper-cube is then incrementally either increased or decreased in size, until the probability ratio of contained samples matches the threshold t within some tolerance, or until it contains more than 1% of the total samples.

(3) The faces of the hyper-cube are then iteratively adjusted (expand or contract), to adapt to the density of the contained samples, while enforcing the condition $f_{\max}/f_{\min} \leq t$. This step turns the D-dimensional hyper-cube into a D-dimensional hyper-rectangle. This 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 $f_{\max}/f_{\min} > t$.

Figure 4(d) shows the resulting set of M hyper-rectangles for our example.

An in-depth description of the algorithm can be found in Ref. 19 and an open-source implementation of it is available as part of the BAT.jl software package.[20]

3.3. *Integral estimates*

Once M integration regions are defined, we can compute the integral estimates $\hat{I}_i^A$ for each Δ_i^B, according to Eq. (12). The procedure is the same for $\hat{I}_i^B$, so we shall drop the superscripts A and B in the following two subsections. The two resulting, separate estimates $\hat{I}^A$ and $\hat{I}^B$ will then be combined in Subsec. 3.5 to obtain the final estimate.

From the distribution of all estimates $\hat{I}_i$ we select only the 68% central percentile to reject outliers — a procedure that was empirically found to work well. This is indicated in Fig. 4(d) labeled as "accepted" and "rejected" rectangles. We proceed to combine the remaining estimates $\hat{I}_i$ into a single estimate $\hat{I}$ using a robust and unbiased estimator for the combination of correlated measurements as suggested in Ref. 17

$$\hat{I} = \sum_i w_i \hat{I}_i\,, \quad \sigma^2(\hat{I}) = \sum_{i,j} w_i w_j \bar{\sigma}_{ij}\,, \tag{19}$$

where the weights w_i are defined as:

$$w_i = \frac{\frac{1}{\bar{\sigma}_i^2}}{\sum_j \frac{1}{\bar{\sigma}_j^2}} \,. \tag{20}$$

The variances $\bar{\sigma}_i^2 \equiv \bar{\sigma}_{ii}$ and covariances $\bar{\sigma}_{ij}$ of the mean assigned to integration regions Δ_i and Δ_j are estimated in the next section.

3.4. *Covariance estimate*

The following procedure is used to estimate the covariance between individual integral estimates $\hat{I}_i$:

(1) We partition $\{\Lambda\}$ into a number S of subsets $\{\Lambda_1\}$, $\{\Lambda_2\} \cdots \{\Lambda_S\}$, chosen in a way that reduces their correlation. The default value for S is 10.

(2) Separate estimates $\hat{I}_{i,k}$ (k enumerates the S partitions) of the integral are then performed for all sample subsets $\{\Lambda_k\}$ resulting in S integral estimates for each subspace Δ_i:

$$\left[\hat{I}_{i,1}\ \hat{I}_{i,2} \cdots\ \hat{I}_{i,S}\right] .$$

(3) The covariance of the separate integral values then is

$$\sigma_{ij}^2 = \frac{1}{S-1} \sum_{k=1}^{S} (\hat{I}_{i,k} - \bar{I}_i)(\hat{I}_{j,k} - \bar{I}_j)$$

with

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

(4) The estimate of the covariance of $\bar{I}_i \approx \hat{I}_i$ and $\bar{I}_j \approx \hat{I}_j$ is then $\bar{\sigma}_{ij}^2 = \frac{\sigma_{ij}^2}{S}$.

3.4.1. *Alternative covariance estimation method*

As S cannot be large without requiring a very large number of samples, the entries of $\bar{\sigma}_{ij}$ tend to be noisy. In practice, we have found this to be tolerable. However, we have developed an optional alternative procedure to estimate $\bar{\sigma}_{ij}$, for use cases where the total number of samples does not allow for a sufficient number of subsets.

The diagonal elements $\bar{\sigma}_{ii}^2$ can be estimated based on the theoretical uncertainties derived in Subsec. 2.2, using the effective number of samples (estimated from the autocorrelation of the samples).

The off-diagonal elements $\bar{\sigma}_{i\neq j}$ are estimated heuristically, based on the overlap of the hyper-rectangles (quantified as the total weight of the samples they share):

$$\bar{\sigma}_{i\neq j} = \rho_{i\neq j} \bar{\sigma}_i \bar{\sigma}_j \,,$$

where

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

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

Some details of this alternative procedure are still under development, so we do not present an in-depth description and formal arguments for it is validity here. We have observed close agreement with the empirical covariance (step 3) in many cases and hope to further refine this method in the future.

3.5. *Final AHMI integral estimate*

As we have now obtained two values of $\hat{I}$ and two variances $\sigma^2(\hat{I})$ from the two sets $\{\Lambda^A\}$ and $\{\Lambda^B\}$, we combine those into the final result like

$$\hat{I} = \frac{\hat{I}^A/\sigma_A^2 + \hat{I}^B/\sigma_B^2}{1/\sigma_A^2 + 1/\sigma_B^2}$$

with variance estimate

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

4. Benchmark Examples

To validate our algorithm, we apply it to estimate the integral of several test functions in varying dimensionality (up to $d = 25$) for which an analytic (or accurate numerical) solution of the integral value is available.

The test functions were chosen to pose different challenges to the algorithm. For our first test problem, we start with the canonical example of a multivariate normal distribution. For the second test case we look at Gaussian shells, for which the mode of the distribution does not lie on a single point but has infinite modes on $(d-1)$-dimensional surface. Next we study the heavy-tailed Cauchy distribution with multiple modes, and in the end explore the asymmetric “Funnel” function. Additional information of the number of integration volumes used and the computation time are only provided for the first example, as these are very similar for the other three examples.

The samples $\{\Lambda\}$ on which our integration is based are obtained from Metropolis–Hastings MCMC,[20] and in the case of the multivariate normal from *i.i.d.* sampling. The sample size is fixed to $2 \cdot 10^6$ for Gaussian shells example and it is equal to 10^6 for other examples.

4.1. *Multivariate normal distribution*

The first test case is a unit normal distribution (centered at zero, width one) in two up to 25 dimensions. The samples input to AHMI are obtained from *i.i.d.* sampling. The resulting integral estimates are shown in Fig. 5 as a function of the dimensionality for three different threshold values $t = [100, 500, 1000]$.

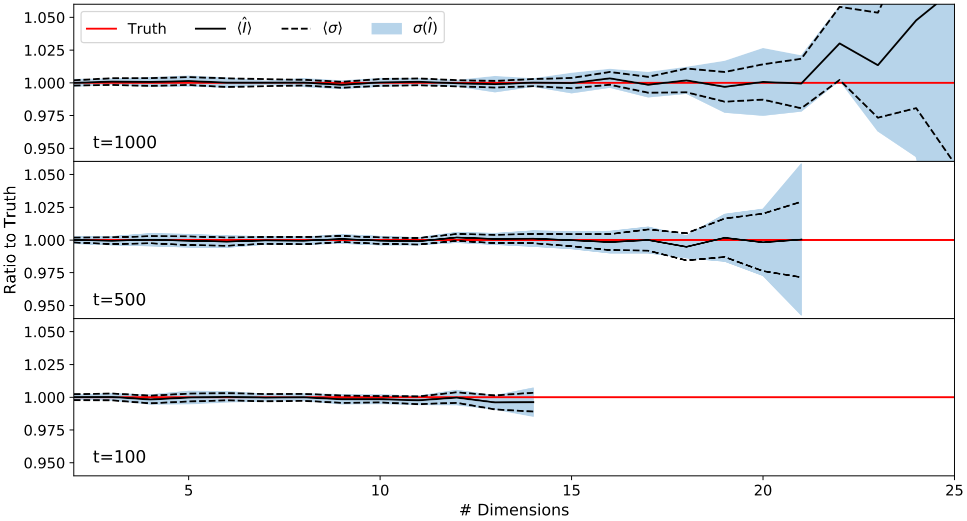


Fig. 5. AHMI value for the multivariate normal distribution as ratio to the true integral shown as a function of dimensionality for three threshold values $t = [100, 500, 1000]$. The solid black lines give the mean result over ten independent trials and the shaded bands show the standard deviation of these trials. The dashed lines show the average errors reported by AHMI. Samples are obtained from *i.i.d.* sampling, 10^6 for each run.

For both threshold values, $t = [500, 1000]$, we get unbiased and consistent results up to around 21 dimensions, after which results start to become positively biased for $t = 1000$. This bias seems to be intrinsic to the method itself or our implementation of the algorithm and will be the subject of future studies. The coverage for one standard error for $t = 500$ (220 trials) is 0.52 ± 0.03. For the threshold value $t = 100$ we get an unbiased integral estimate up to 14 dimensions, and after that

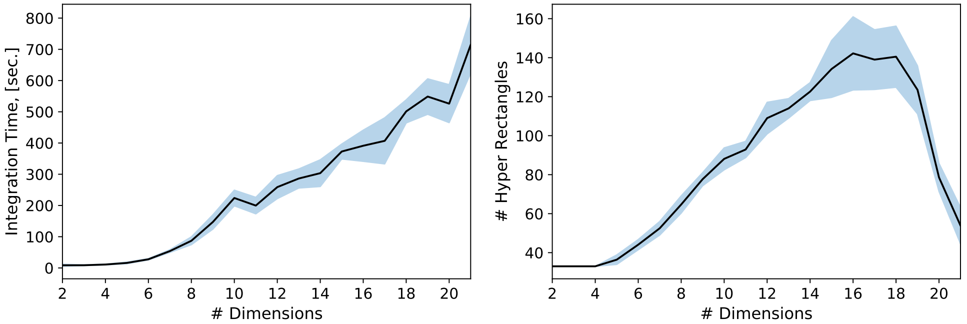


Fig. 6. Left: Average AHMI execution time for the multivariate normal distribution ($t = 500$) in total CPU seconds (run on a system with a 2.3 GHz Intel Xeon 6140 processor, SPECrate2017-FP equivalent ca. 180) as a function of dimensionality. Right: Average number of hyper-rectangles used by AHMI for computing the integral estimate for the multivariate normal distribution as a function of dimensionality.

hyper-rectangles can no longer be created. The default value of $t = 500$ was used for the remaining examples below.

In Fig. 6 we show the execution time of the algorithm, and the number of hyper-rectangles used for integration for the threshold value $t = 500$. The execution time rises with the number of dimensions almost linearly. The change in slope at low dimensionality is likely due to CPU caching behavior. The number of hyper-rectangles starts to decay after 18 dimensions indicating that there exist fewer hyper-rectangles that satisfy Eq. (18).

4.2. *Gaussian shell distribution*

The functional form was given in Eq. (17) and an example distribution in the first two dimensions is shown in Fig. 7. The AHMI algorithm results (Fig. 8) show a

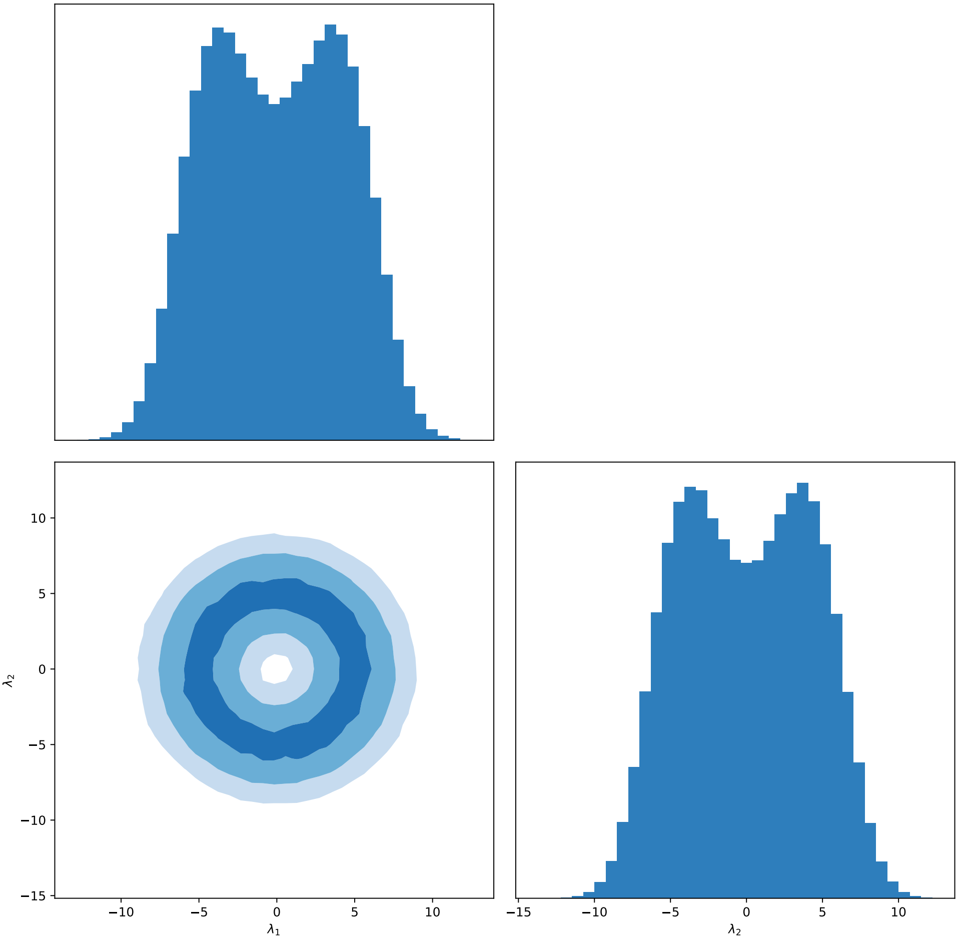


Fig. 7. One- and two-dimensional distributions of samples along the first two dimensions of the Gaussian shell target function.

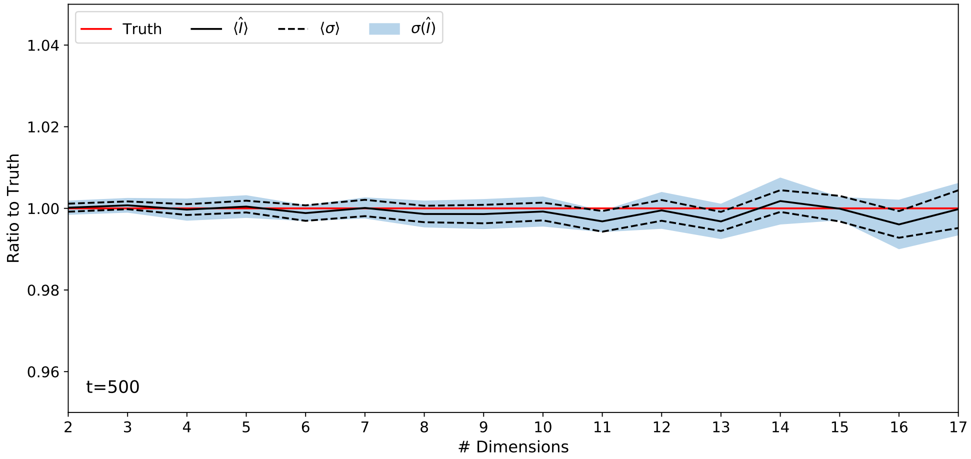


Fig. 8. AHMI value for the Gaussian shell distribution as ratio to the true integral shown as a function of dimensionality. The solid line gives the mean result over ten independent trials and the shaded band show the standard deviation of these trials. The dashed lines show the average errors reported by AHMI. The samples are obtained from Metropolis–Hastings MCMC and the sample size is $2 \cdot 10^6$.

similar behavior as for the multivariate normal distribution for this more complicated test function. However, integration was possible up to 17 dimensions. Up to that point the integral estimates, including errors, are well behaved, with a coverage for one standard error of 0.48 ± 0.04 (160 trials).

4.3. *Multimodal Cauchy distribution*

The Cauchy distribution, with its heavy tails, is a notoriously difficult problem and used here to point out possible weaknesses of our algorithm. We further increase complexity for the hyper-volume creation process by using four separate, shifted Cauchy distributions creating multiple modes. The functional form can be written as

$$f(\lambda) = \prod_{i=1}^{2} \frac{1}{2}[\text{Cauchy}(\lambda_i|\mu,\sigma) + \text{Cauchy}(\lambda_i|-\mu,\sigma)] \times \prod_{j=3}^{n} \text{Cauchy}(\lambda_j|0,\sigma)\,, \tag{21}$$

where $\mu = 1$, $\sigma = 0.2$ and n is a dimensionality of λ. An example of this target distributions is provided in Fig. 9. The integration region extends from $[-8, 8]$ in each dimension. Our results are collected in Fig. 10 and indicate that integration is possible up to seven dimensions, given the fixed sample size of 10^6. For this range, the AHMI results are very reliable. The coverage (120 trials, one standard error) for this function is 0.42 ± 0.05.

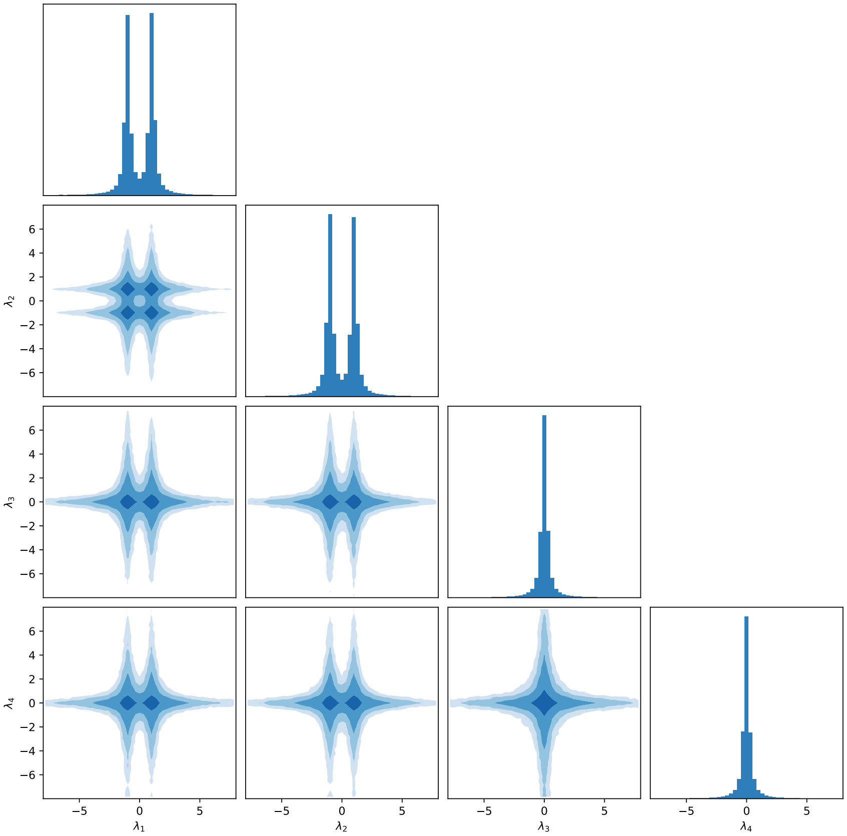


Fig. 9. One- and two-dimensional distributions of samples along the first four dimensions of the multimodal Cauchy target function.

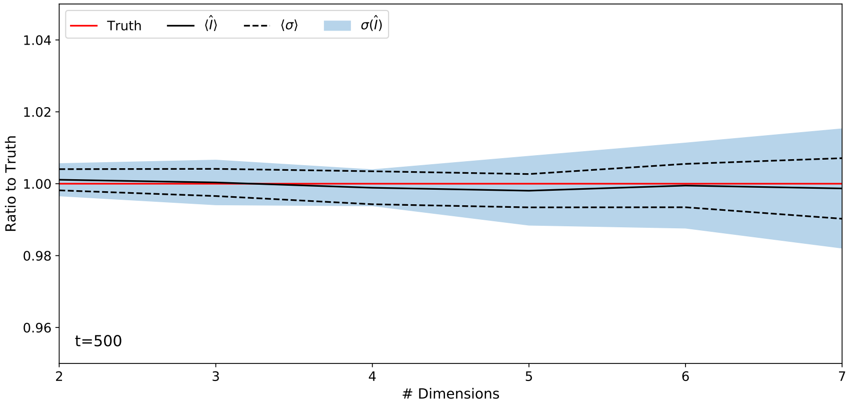


Fig. 10. AHMI value for the multimodal Cauchy distribution as ratio to the true integral shown as a function of dimensionality. The solid line gives the mean result over twenty independent trials and the shaded band show the standard deviation of these trials. The dashed lines show the average errors reported by AHMI.

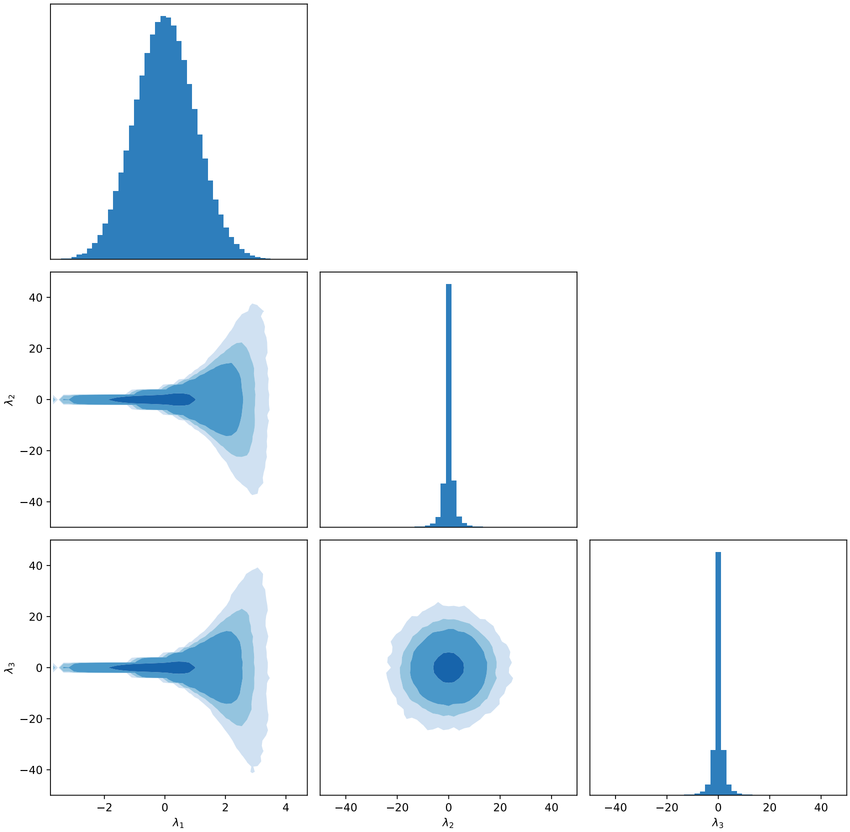


Fig. 11. One- and two-dimensional distributions of samples along the first three dimensions of the Funnel target function.

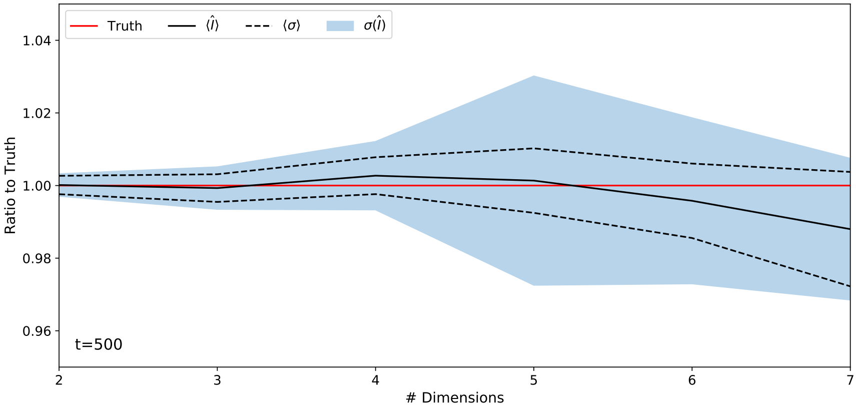


Fig. 12. AHMI value for the "Funnel" distribution as ratio to the true integral shown as a function of dimensionality. The solid line gives the mean result over twenty independent trials, with the shaded band showing the standard deviation of these trials. The dashed lines show the average errors reported by AHMI.

4.4. *Funnel distribution*

The final problem we study is the so-called "Funnel" distribution, that is described in Ref. 21. The functional form of this distribution can be written as

$$f(\lambda) = \mathcal{N}(\lambda_1|0, a^2) \prod_{i=2}^{n} \mathcal{N}(\lambda_i|0, \exp(2b\lambda_1))\,, \tag{22}$$

where $a = 1$, $b = 0.5$ and n is a dimensionality of λ. An example of the distribution in its first three dimensions on the parameter range $[-50, 50]$ is provided in Fig. 11. The results (Fig. 12) show a similar performance as for the previous example with reliable estimates up to seven dimensions, with a coverage of 0.41 ± 0.04 (120 trials, one standard error).

5. Conclusion

We have developed an Adaptive Harmonic Mean Integration (AHMI) algorithm that can be used to integrate a non-normalized density function using the samples drawn according to the probability distribution proportional to the function. The fundamental assumption is that the sampling algorithm has faithfully produced samples from this distribution. Given this, the AHMI algorithm can be used to produce both an estimate of the integral of the function over its full support as well as an estimate of the uncertainty of the integral. In this first implementation of the AHMI algorithm, finite hyper-rectangles are generated in the whitened space of the samples covering the full support. The adaptive algorithm ensures that the range of function values enclosed by the hyper-rectangles is limited such that the variance of the integral results are moderate. This allows for reliable results both for the integral values as well as for reliable uncertainty estimates.

The algorithm has been tested on a number of examples and found to produce reliable and unbiased integral estimates up to around 20 dimensions for the first two test problems, and up to seven for the latter two, more difficult test cases. The reported errors provide a useful measure of uncertainty, while slightly under covering at around 40–50% (expecting 68%). The use of hyper-rectangles however limits the applicability to a not-too-large number of dimensions (≈ 20 in the case of the multivariate normal distribution) because a large fraction of the volume is in the corners of the hyper-rectangles. In cases where we have a unimodal distribution, we anticipate that regions defined as shells between hyper-spheres of different radii would be more appropriate. We leave this for a future development of the algorithm.

Acknowledgments

This research was supported by the European Union's Framework Programme for Research and Innovation Horizon 2020 (2014–2020) under the Marie Sklodowska-Curie Grant Agreement No. 765710 and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy (EXC-2094) 390783311.

References

1. C. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd edn. (Springer, 2004).
2. H. Jeffreys, *Theory of Probability*, 3rd edn. (Clarendon Press, Oxford, 1961).
3. E. T. Jaynes, *Probability Theory: The Logic of Science* (Cambridge University Press, Cambridge, 2003).
4. M. A. Newton and A. E. Raftery, *J. R. Stat. Soc. Ser. B* **56**, 3 (1994).
5. M. D. Weinberg, *Bayesian Anal.* **7**, 737 (2012).
6. D. N. VanDerwerken and S. C. Schmidler, Parallel Markov chain Monte Carlo, arXiv:1312.7479v1.
7. A. Caldwell and C. Liu, Target density normalization for a Markov chain Monte Carlo algorithm, arXiv:1410.7149v2.
8. N. Friel and J. Wyse, *Stat. Neerl.* **66**, 2800 (2012).
9. L. Tierney and J. B. Kadane, *J. Amer. Stat. Assoc.* **81**, 82 (1986).
10. S. Chib and I. Jeliazkov, *J. Amer. Stat. Assoc.* **96**, 270 (2001).
11. R. Neal, *J. R. Stat. Soc. Ser. B* **56**, 41 (1994).
12. C. P. Robert and D. Wraith, *AIP Conf. Proc.* **1193**, 251 (2009).
13. J. Skilling, *Bayesian Anal.* **1**, 833 (2006).
14. A. Gelman and X. L. Meng, *Stat. Sci.* **13**, 163 (1998).
15. N. Friel and A. N. Pettitt, *J. R. Stat. Soc. Ser. B* **70**, 589 (2008).
16. https://radfordneal.wordpress.com/2008/08/17/the-harmonic-mean-of-the-likelihood-worst-monte-carlo-method-ever/.
17. M. Schmelling, *Phys. Scr.* **51**, 676 (1995).
18. T. Hahn, *Comput. Phys. Commun.* **176**, 712 (2007).
19. R. Schick, Adaptive harmonic mean integration technique and its application to neutrino physics, MPP-2018-435, https://publications.mppmu.mpg.de/?action=search&mpi=MPP-2018-435.
20. O. Schulz *et al.*, BAT.jl — A Julia-based tool for Bayesian inference, arXiv:2008.03132.
21. H. Jia and U. Seljak, Normalizing constant estimation with Gaussianized bridge sampling, arXiv:1912.06073 [stat.ML].