Fast Automatic Bayesian Cubature using Matching Kernels and Designs

Jagadeeswaran R.
Adviser: Prof. Fred J Hickernell

Department of Applied Mathematics, Illinois Institute of Technology
jrathin1@iit.edu

Thesis Defense • Oct 22, 2019

Contents

Numerical Integration

A fundamental problem in various fields, including finance, machine learning and statistics,

$$\mu = \int_{\mathbb{R}^d} g(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mathbb{E}[f(\boldsymbol{X})], \quad \text{where} \quad \boldsymbol{X} \sim \mathcal{U}[0,1]^d \tag{1}$$

by a cubature rule $\hat{\mu}_n := w_0 + \sum_{j=1}^{n} f(\boldsymbol{x}_j) w_j$

using points $\{\boldsymbol{x}_j\}_{j=1}^n$ and associated weights w_j.

The goal of this work is to

- Develop an automatic algorithm for integration
- Use an extensible point-set and an algorithm that allows extending points
- Determine n such that, given ϵ, $|\mu - \hat{\mu}_n| \leqslant \epsilon$

Bayesian Cubature

The Bayesian approach for numerical analysis was popularized by Diaconis (1988). Diaconis interpreted the most well known numerical methods, such as 1) trapezoidal rule and 2) splines, from the statistical point of view.

Bayesian approach for numerical integration that is known as Bayesian cubature as introduced by O'Hagan (1991).

- Assume f is drawn from a Gaussian process
  - Need to estimate the mean and Covariance kernel
- Parameter estimation (Empirical, Cross validation) is expensive in general
  - Use points and kernel for which it is cheap

Problem:

- How to choose $\{x_i\}_{i=1}^n$, and $\{w_i\}_{i=1}^n$ to make $|\mu - \hat{\mu}_n|$ small? what is err_{CI}? (Bayesian posterior error)
- How to find n such that $|\mu - \hat{\mu}_n| \leqslant \text{err}_{\text{CI}} \leqslant \epsilon$? (automatic cubature)

Bayesian posterior error

Assume random $f \sim \mathcal{GP}(m, s^2 C_\theta)$, a Gaussian process with mean m and covariance kernel $s^2 C_\theta$, $C_\theta : [0,1] \times [0,1] \to \mathbb{R}$.

$$\text{Define} \quad c_0 = \int_{[0,1]\times[0,1]} C_\theta(\boldsymbol{x}, \boldsymbol{t}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{t},$$

$$\boldsymbol{c} = \left(\int_{[0,1]} C_\theta(\boldsymbol{x}_i, \boldsymbol{t}) \mathrm{d}\boldsymbol{t} \right)_{i=1}^{n}, \quad \mathsf{C}_\theta = \left(C_\theta(\boldsymbol{x}_i, \boldsymbol{x}_j) \right)_{i,j=1}^{n}$$

$$\mu - \hat{\mu}_n \big| \boldsymbol{y} \sim \mathcal{N}\left(0, \quad s^2(c_0 - \boldsymbol{c}^T \mathsf{C}_\theta^{-1} \boldsymbol{c})\right) \quad \text{where } \boldsymbol{y} = \left(f(\boldsymbol{x}_i)\right)_{i=1}^{n}.$$

Choosing $\hat{\mu}_n = \underbrace{m(1 - \mathbf{1}^T \mathsf{C}_\theta^{-1} \boldsymbol{c})}_{w_0} + (\underbrace{\mathsf{C}_\theta^{-1} \boldsymbol{c}}_{\boldsymbol{w}})^T \boldsymbol{y}$ makes error unbiased.

If $m = 0$, fixed, choosing $\boldsymbol{w} = \mathsf{C}_\theta^{-1} \boldsymbol{c}$ makes error unbiased. Please note m, s and θ needs to be inferred.

Diaconis (1988), O'Hagan (1991), Ritter (2000), Rasmussen and Williams (2003), Briol et al. (2018+), Traub et al. (1988) and others

Parameter estimation - Empirical Bayes

Maximize the log-likelihood of the parameters given the data $\boldsymbol{y} = (f(\boldsymbol{x}_i))_{i=1}^n$ w.r.t. m then s^2, then with θ:

$$m_{\mathrm{EB}} = \frac{\mathbf{1}^T \mathsf{C}_\theta^{-1} \boldsymbol{y}}{\mathbf{1}^T \mathsf{C}_\theta^{-1} \mathbf{1}}, \qquad \text{(Explicit)}$$

$$s^2_{\mathrm{EB}} = \frac{1}{n} \boldsymbol{y}^T \left[\mathsf{C}_\theta^{-1} - \frac{\mathsf{C}_\theta^{-1} \mathbf{1} \mathbf{1}^T \mathsf{C}_\theta^{-1}}{\mathbf{1}^T \mathsf{C}_\theta^{-1} \mathbf{1}} \right] \boldsymbol{y}, \qquad \text{(Explicit)}$$

$$\theta_{\mathrm{EB}} = \operatorname*{argmin}_\theta \log \left(\frac{1}{2n} \log(\det \mathsf{C}_\theta) + \log(s_{\mathrm{EB}}) \right) \qquad \text{(numeric)}$$

$$\hat{\mu}_{\mathrm{EB}} = \left(\frac{(1 - \mathbf{1}^T \mathsf{C}_\theta^{-1} \boldsymbol{c}) \mathbf{1}}{\mathbf{1}^T \mathsf{C}_\theta^{-1} \mathbf{1}} + \boldsymbol{c} \right)^T \mathsf{C}_\theta^{-1} \boldsymbol{y}, \qquad \text{(Explicit)}$$

The credible interval width, $\mathrm{err}_{\mathrm{EB}}$, is given by

$$\mathrm{err}_{\mathrm{EB}} = 2.58 \, s_{\mathrm{EB}} \sqrt{c_0 - \boldsymbol{c}^T \mathsf{C}_\theta^{-1} \boldsymbol{c}}$$

1: **procedure** AUTOBAYESCUBATURE(f, ϵ)

Require: a generator for the sequence $x_1, x_2, \ldots$; a black-box function, f; an absolute error tolerance, $\varepsilon > 0$; the positive initial sample size, n_0; the maximum sample size n_{max}

2: $n \leftarrow n_0,\ n' \leftarrow 0,\ \text{err}_\text{n} \leftarrow \infty$

3: **while** $\text{err}_{\text{CI}} > \varepsilon$ and $n \leqslant n_{\text{max}}$ **do**

4: Generate $\{x_i\}_{i=n'+1}^{n}$ and sample $\{f(x_i)\}_{i=n'+1}^{n}$,

5: Compute parameters, compute error bound err_{CI}

6: $n' \leftarrow n,\ n \leftarrow 2 \times n'$

7: **end while**

8: Sample size to compute $\hat{\mu}$, $n \leftarrow n'$

9: Compute approximate $\hat{\mu}_n$, the approximate integral

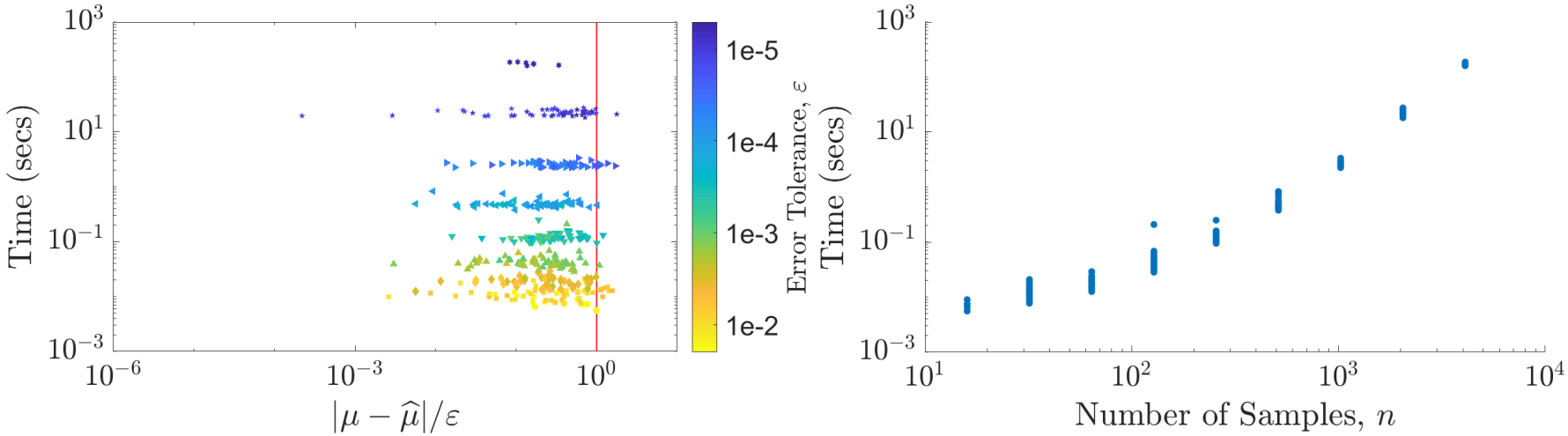
10: **return** $\hat{\mu}_n$ ▷ Integral estimate $\hat{\mu}_n$

11: **end procedure**

Multivariate Gaussian integration with Matérn kernel

Matérn covariance kernel: $C_\theta(\boldsymbol{x}, \boldsymbol{t}) = \prod_{\ell=1}^{d} \exp(-\theta|\boldsymbol{x}_\ell - \boldsymbol{t}_\ell|)(1 + \theta|\boldsymbol{x}_\ell - \boldsymbol{t}_\ell|)$ (2)

$$d = 3, \quad \boldsymbol{a} = \begin{pmatrix} -6 \\ -2 \\ -2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 5 \\ 2 \\ 1 \end{pmatrix}, \quad \mathsf{L} = \begin{pmatrix} 4 & 1 & 1 \\ 0 & 1 & 0.5 \\ 0 & 0 & 0.25 \end{pmatrix}.$$



Problem: Computation time (in seconds) increases rapidly, so it is not practical to use more than 4000 points in the cubature.

Fast Bayesian Transform

Choose the kernel C_θ and $\{\boldsymbol{x}_i\}_{i=1}^n$, so the Gram matrix $\mathsf{C}_\theta = \left(C_\theta(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)_{i,j=1}^n$ has:

$$\mathsf{C}_\theta = (\boldsymbol{C}_1, ..., \boldsymbol{C}_n) = \frac{1}{n}\mathsf{V}\Lambda\mathsf{V}^H, \quad \mathsf{V} = (\boldsymbol{V}_1, ..., \boldsymbol{V}_n) = (\boldsymbol{v}_1, ..., \boldsymbol{v}_n)^T$$

C_θ is a fast Bayesian transform kernel, if

$$\mathsf{V} \text{ may be identified analytically,} \tag{3a}$$

$$\boldsymbol{v}_1 = \boldsymbol{V}_1 = \mathbf{1}, \tag{3b}$$

$$\text{Computing } \widetilde{\boldsymbol{b}} = \mathsf{V}^H\boldsymbol{b} \text{ requires only } \mathcal{O}(n \log n) \text{ operations } \forall \boldsymbol{b}. \tag{3c}$$

The covariance kernel may also be normalized

$$\int_{[0,1]^d} C_\theta(\boldsymbol{t}, \boldsymbol{x})\, \mathrm{d}\boldsymbol{t} = 1 \qquad \forall \boldsymbol{x} \in [0,1]^d, \text{ leading to } c_0 = 1 \text{ and } \boldsymbol{c} = \mathbf{1}. \tag{4}$$

Please note that

$$\Lambda = \mathrm{diag}(\boldsymbol{\lambda}), \quad \boldsymbol{\lambda} = (\lambda_1, ..., \lambda_n) = \mathsf{V}^H\boldsymbol{C}_1.$$

Parameter estimation - Full Bayes

Treat m and s as hyper-parameters with a non-informative, conjugate prior, namely $\rho_{m,s^2}(\xi,\lambda) \propto 1/\lambda$. Then the posterior density for the integral μ given the data is

$$\rho_\mu(z|\boldsymbol{f}=\boldsymbol{y}) \propto \int_0^\infty \int_{-\infty}^\infty \rho_\mu(z|\boldsymbol{f}=\boldsymbol{y}, m=\xi, s^2=\lambda)\rho_f(\boldsymbol{y}|\xi,\lambda)\rho_{m,s^2}(\xi,\lambda)\,\mathrm{d}\xi\mathrm{d}\lambda$$

$$\propto \left(1+\frac{1}{n-1}\frac{(z-\mu_{\text{full}})^2}{\hat{\sigma}^2_{\text{full}}}\right)^{-n/2}$$

Where

$$\mu_{\text{full}} = \mu_{\text{EB}}$$

$$\hat{\sigma}^2_{\text{full}} = \frac{1}{n-1}\boldsymbol{y}^T\left[C_\theta^{-1} - \frac{C_\theta^{-1}\mathbf{1}\mathbf{1}^T C_\theta^{-1}}{\mathbf{1}^T C_\theta^{-1}\mathbf{1}}\right]\boldsymbol{y} \times \left[\frac{(1-\boldsymbol{c}^T C_\theta^{-1}\mathbf{1})^2}{\mathbf{1}^T C_\theta^{-1}\mathbf{1}} + (c_0 - \boldsymbol{c}^T C_\theta^{-1}\boldsymbol{c})\right]$$

$$\mathbb{P}_f\left[|\mu - \hat{\mu}_{\text{full}}| \leqslant \text{err}_{\text{full}}\right] = 99\%,$$

$$\text{err}_{\text{full}} := t_{n-1,0.995}\hat{\sigma}_{\text{full}} > \text{err}_{\text{EB}}$$

Parameter estimation - Generalized Cross validation

Let $\widetilde{y}_i = \mathbb{E}[f(\boldsymbol{x}_i)|\boldsymbol{f}_{-i} = \boldsymbol{y}_{-i}]$. The cross-validation criterion, which is to be minimized, is sum of squares of the difference between these conditional expectations and the observed values: :

$$\mathrm{CV} = \sum_{i=1}^{n} (y_i - \widetilde{y}_i)^2 = \sum_{i=1}^{n} \left(\frac{\zeta_i}{a_{ii}} \right)^2, \quad \text{where } \boldsymbol{\zeta} = \mathsf{C}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{y} - m\mathbf{1}),$$

$$a_{ii} \text{ are diagonal elems of } \mathsf{C}_{\boldsymbol{\theta}}^{-1} = \begin{pmatrix} a_{ii} & \boldsymbol{A}_{-i,i}^T \\ \boldsymbol{A}_{-i,i} & \mathsf{A}_{-i,-i} \end{pmatrix}$$

$$\mathrm{GCV} = \frac{\sum_{i=1}^{n} \zeta_i^2}{\left(\frac{1}{n} \sum_{i=1}^{n} a_{ii} \right)^2} = \frac{(\boldsymbol{y} - m\mathbf{1})^T \mathsf{C}_{\boldsymbol{\theta}}^{-2} (\boldsymbol{y} - m\mathbf{1})}{\left(\frac{1}{n} \operatorname{trace}(\mathsf{C}_{\boldsymbol{\theta}}^{-1}) \right)^2}.$$

$$\boldsymbol{\theta}_{\mathrm{GCV}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \left\{ \log \left(\boldsymbol{y}^T \left[\mathsf{C}_{\boldsymbol{\theta}}^{-2} - \frac{\mathsf{C}_{\boldsymbol{\theta}}^{-2} \mathbf{1}\mathbf{1}^T \mathsf{C}_{\boldsymbol{\theta}}^{-2}}{\mathbf{1}^T \mathsf{C}_{\boldsymbol{\theta}}^{-2} \mathbf{1}} \right] \boldsymbol{y} \right) - 2 \log \left(\operatorname{trace}(\mathsf{C}_{\boldsymbol{\theta}}^{-1}) \right) \right\}$$

The credible interval width, $\mathrm{err}_{\mathrm{EB}}$, is given by

$$\mathrm{err}_{\mathrm{GCV}} = 2.58 \, s_{\mathrm{GCV}} \sqrt{c_0 - \boldsymbol{c}^T \mathsf{C}_{\boldsymbol{\theta}}^{-1} \boldsymbol{c}}$$

Computing the θ, err_{CI}, and $\hat{\mu}$ faster

$$\theta_{\text{EB}} = \underset{\theta}{\text{argmin}} \left[\log \left(\sum_{i=2}^{n} \frac{|\tilde{y}_i|^2}{\lambda_i} \right) + \frac{1}{n} \sum_{i=1}^{n} \log(\lambda_i) \right] \tag{5a}$$

$$\theta_{\text{GCV}} = \underset{\theta}{\text{argmin}} \left[\log \left(\sum_{i=2}^{n} \frac{|\tilde{y}_i|^2}{\lambda_i^2} \right) - 2 \log \left(\sum_{i=1}^{n} \frac{1}{\lambda_i} \right) \right] \tag{5b}$$

$$\text{err}_{\text{EB}} = \frac{2.58}{n} \left\{ \sum_{i=2}^{n} \frac{|\tilde{y}_i|^2}{\lambda_i} \left(1 - \frac{n}{\lambda_1} \right) \right\}^{1/2} \tag{6a}$$

$$\text{err}_{\text{full}} = t_{n-1,0.995} \left\{ \frac{1}{n(n-1)} \sum_{i=2}^{n} \frac{|\tilde{y}_i|^2}{\lambda_i} \left(\frac{\lambda_1}{n} - 1 \right) \right\}^{1/2} \tag{6b}$$

$$\text{err}_{\text{GCV}} = \frac{2.58}{n} \left\{ \sum_{i=2}^{n} \frac{|\tilde{y}_i|^2}{\lambda_i^2} \left[\frac{1}{n} \sum_{i=1}^{n} \frac{1}{\lambda_i} \right]^{-1} \times \left(1 - \frac{n}{\lambda_1} \right) \right\}^{1/2} \tag{6c}$$

Similarly, $\hat{\mu}$ can be computed faster $\hat{\mu}_{\text{EB}} = \hat{\mu}_{\text{full}} = \hat{\mu}_{\text{GCV}} = \dfrac{1}{n} \sum_{i=1}^{n} y_i$

Rank-1 Lattice rules : low discrepancy point set

Given the generating vector $\boldsymbol{h}$, the construction of n - Rank-1 lattice points (Dick and Pillichshammer, 2010) is given by

$$\mathsf{L}_{n,\boldsymbol{h}} := \{\boldsymbol{x}_i := (\boldsymbol{h}\phi(i-1) + \boldsymbol{\Delta}) \mod 1; \quad i = 1, \ldots, n\} \tag{7}$$

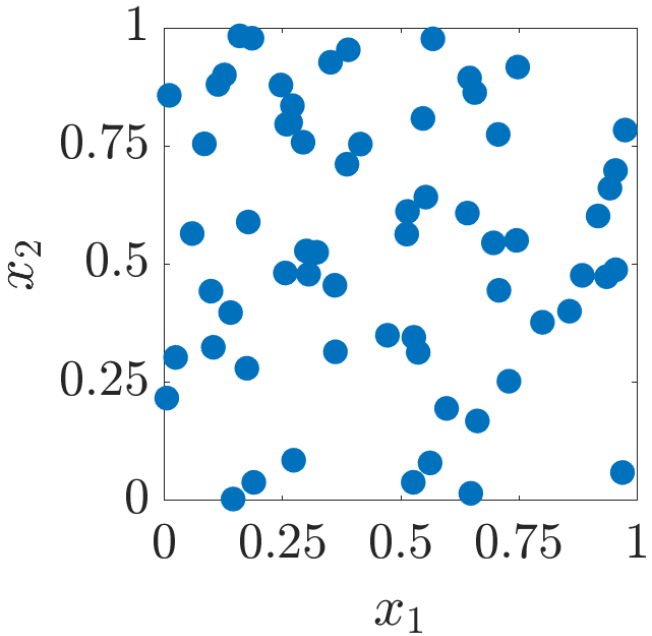
where Δ is a random shift and $\boldsymbol{h}$ is a *generalized Mahler integer* (∞ digit expression) (Hickernell and Niederreiter, 2003) also called generating vector. $\phi(i)$ is the Van der Corput sequence in base 2. Then the Lattice rule approximation is

$$\frac{1}{n}\sum_{i=1}^{n} f\left(\boldsymbol{x}_i\right).$$

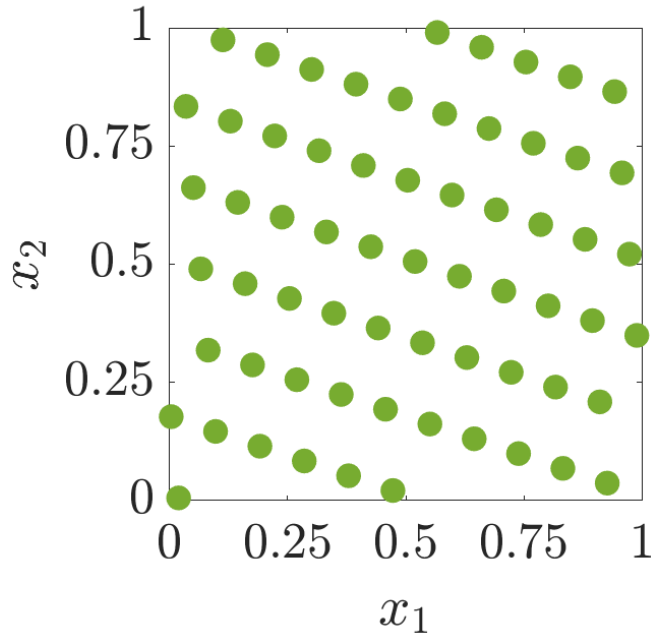
Extensible integration lattices : The number of points in the node set can be increased while retaining the existing points.(Hickernell and Niederreiter, 2003)

Rank-1 Lattice points in $d = 2$

An example of $n = 64$ IID points and Lattice nodes:



(a) IID



(b) Shifted Lattice

Shift invariant kernel + Lattice points = '*Symmetric circulant kernel*' matrix

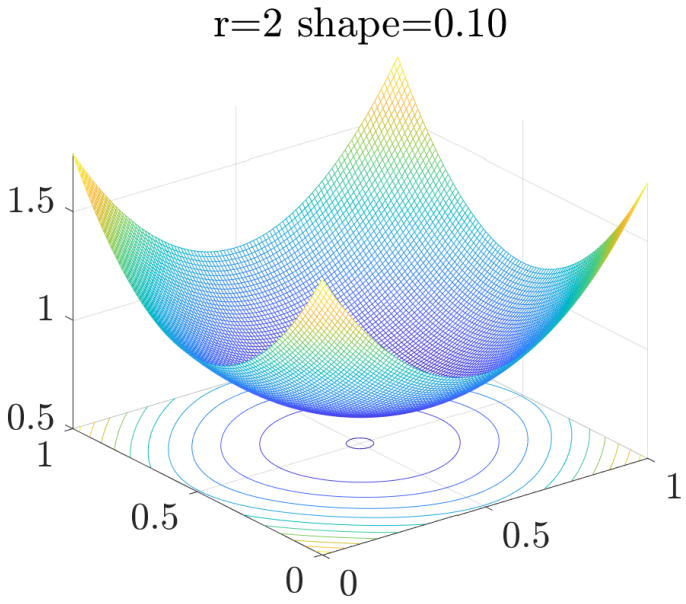
Lattice nodes and shift invariant covariance kernels

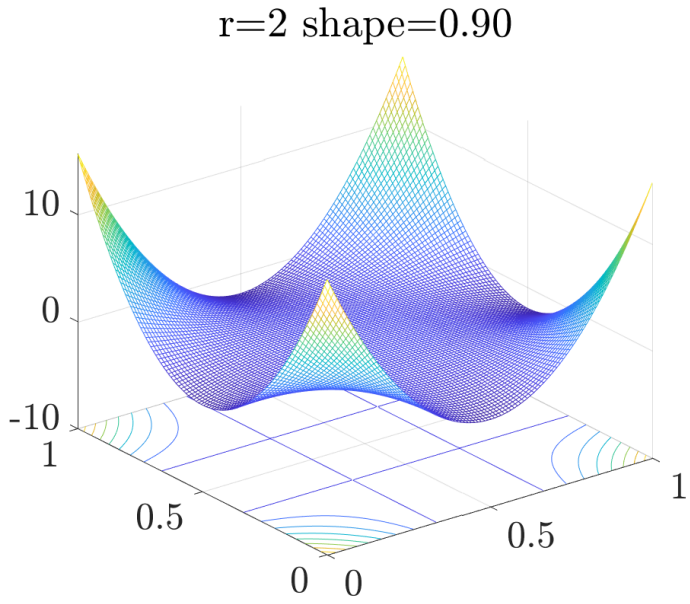
$$C_\theta(\boldsymbol{x},\boldsymbol{t}) = \prod_{\ell=1}^{d}\left[1-\eta_\ell\frac{(2\pi\sqrt{-1})^r}{r!}B_r(|x_\ell - t_\ell|)\right], \quad r\in 2\mathbb{N}, \quad \eta_\ell > 0, \quad \theta = (r,\boldsymbol{\eta})$$

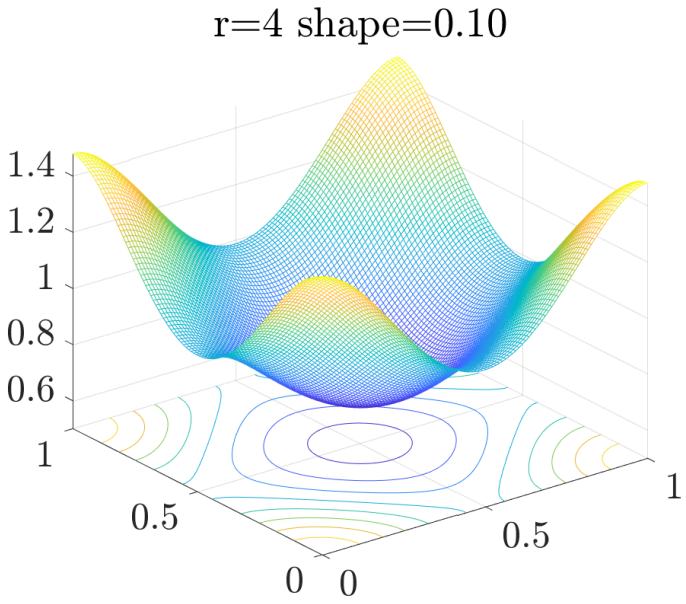
where B_r is Bernoulli polynomial of order r (Olver et al., 2013). We call C_θ, Fourier kernel. Also this kernel satisfies:

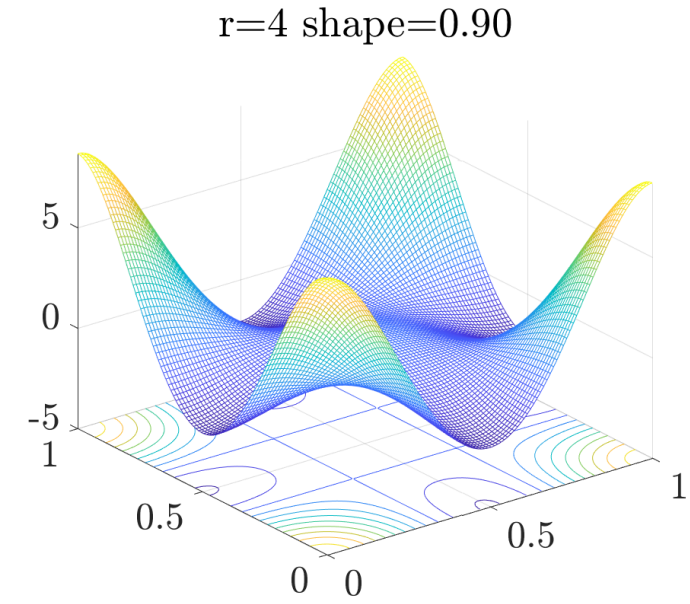
$$c_0 = \int_{[0,1]^2} C_\theta(\boldsymbol{x},\boldsymbol{t})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{t} = 1, \qquad \boldsymbol{c} = \left(\int_{[0,1]} C_\theta(\boldsymbol{x}_i,\boldsymbol{t})\mathrm{d}\boldsymbol{t}\right)_{i=1}^{n} = \mathbf{1}.$$

Fourier kernel



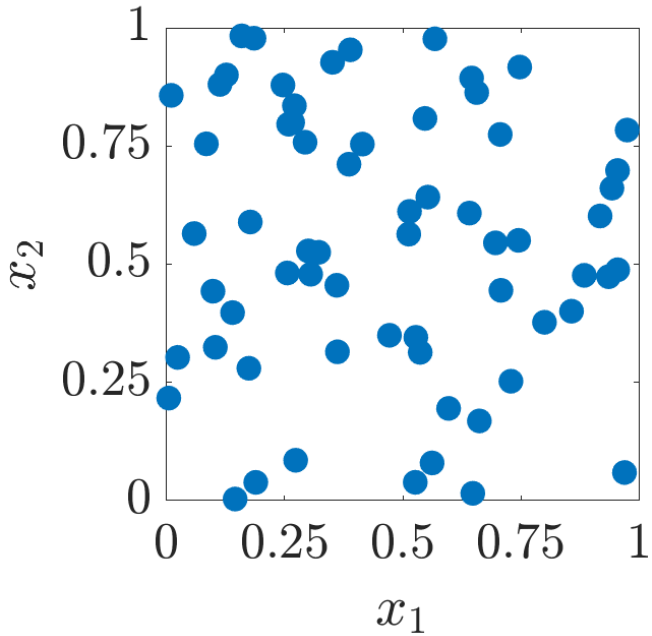




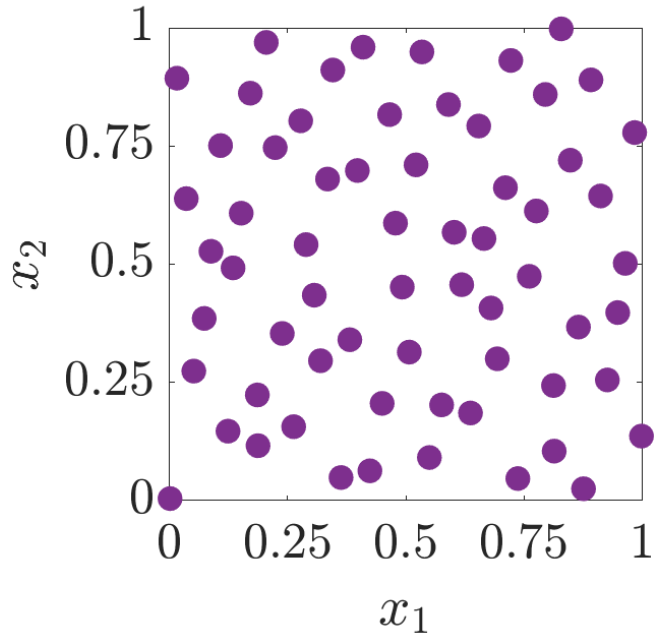


Sobol' Nets

An example of $n = 64$ IID points and Sobol' nets:



(a) IID



(b) Scrambled Sobol'

Walsh Kernels

The Walsh covariance kernels are of the form

$$C_{\theta}(\boldsymbol{x}, \boldsymbol{t}) = K_{\theta}(\boldsymbol{x} \ominus \boldsymbol{t}). \tag{8}$$

where

$$K_{\theta}(\boldsymbol{x} \ominus \boldsymbol{t}) = \prod_{\ell=1}^{d} 1 + \eta_{\ell} \omega_r(x_{\ell} \ominus t_{\ell}), \quad \boldsymbol{\eta} = (\eta_1, \cdots, \eta_d), \quad \theta = (r, \boldsymbol{\eta}) \tag{9}$$

where r is the kernel order, $\boldsymbol{\eta}$ is the kernel shape parameter. For example, explicit expression is available for ω_r in the case of order $r = 1$ (Nuyens, 201308),

$$\omega_1(x) = 6\left(\frac{1}{6} - 2^{\lfloor \log_2 x \rfloor - 1}\right). \tag{10}$$

Walsh Kernels

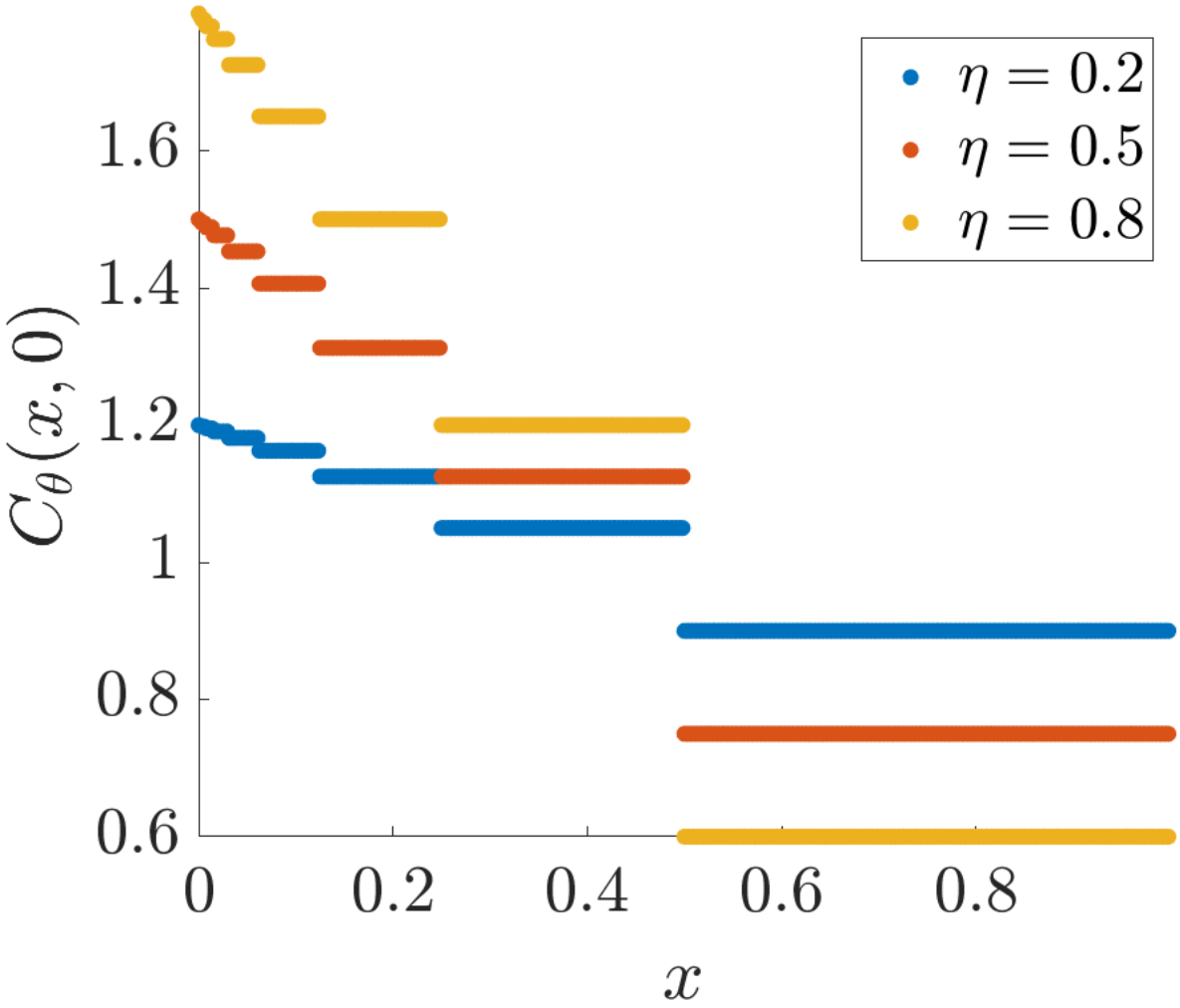


Figure: Walsh kernel of order $r = 1$ in dimension $d = 1$.

Sobol' Nets and Walsh Kernels

Walsh kernels + digital nets = 2×2 *block-Toeplitz* matrix

Fast Bayesian Transform

Theorem

The Walsh-Hadamard matrix $\mathsf{H}^{(m)}$ *factorizes* $\mathsf{C}_{\theta}^{(m)}$*, so that the columns of Walsh-Hadamard matrix are the eigenvectors of* $\mathsf{C}_{\theta}^{(m)}$*, i.e.,*

$$\mathsf{H}^{(m)}\mathsf{C}_{\theta}^{(m)} = \Lambda^{(m)}\mathsf{H}^{(m)}, \quad m \in \mathbb{N},$$

where (m) *denotes the size of the matrix is* $2^m \times 2^m$*.*

By this theorem

$$\mathsf{C}^{(m)} = \frac{1}{n}\mathsf{H}^{(m)}\Lambda^{(m)}\mathsf{H}^{(m)}, \quad \text{where} \quad \mathsf{H}^{(m)} = \underbrace{\mathsf{H}^{(1)} \bigotimes \cdots \bigotimes \mathsf{H}^{(1)}}_{m \text{ times}}. \tag{11}$$

Cancellation error in $\mathrm{err}_{\mathrm{CI}}$

$$\mathrm{err}_{\mathrm{EB}} = 2.58\sqrt{\left(1-\frac{n}{\lambda_1}\right)\frac{1}{n^2}\sum_{i=2}^{n}\frac{|\tilde{y}_i|^2}{\lambda_i}}, \quad \text{may cause cancellation error}$$

Let $\quad C_\theta(\boldsymbol{t},\boldsymbol{x}) = \prod_{\ell=1}^{d}\left[1+\mathring{C}_{\theta,\ell}(t_\ell,x_\ell)\right], \qquad \mathring{C}_{\theta,\ell}:[0,1]\times[0,1]\to\mathbb{R}.$

Direct computation of $\mathring{C}_\theta(\boldsymbol{t},\boldsymbol{x}) = C_\theta(\boldsymbol{t},\boldsymbol{x}) - 1$ introduces cancellation error if the $\mathring{C}_\ell$ are small. So, we employ the iteration,

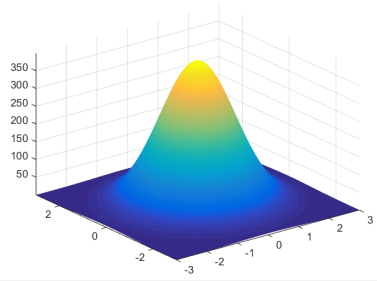
$$\begin{aligned}
\mathring{C}_\theta^{(1)}(\boldsymbol{t},\boldsymbol{x}) &= \mathring{C}_{\theta,1}(t_1,x_1),\\
\mathring{C}_\theta^{(\ell)}(\boldsymbol{t},\boldsymbol{x}) &= \mathring{C}_\theta^{(\ell-1)}[1+\mathring{C}_{\theta,\ell}(t_\ell,x_\ell)] + \mathring{C}_{\theta,\ell}(t_\ell,x_\ell), \qquad \ell = 2,\ldots,d,\\
\mathring{C}_\theta(\boldsymbol{t},\boldsymbol{x}) &= \mathring{C}_\theta^{(d)}(\boldsymbol{t},\boldsymbol{x}).
\end{aligned}$$

Eigenvalues of $\mathring{C}_\theta$: $(\mathring{\lambda}_i)_{i=1}^n = V^T\mathring{C}_1, \quad \mathring{\lambda}_1 = \lambda_1 - n, \lambda_2, \ldots, \lambda_n$

$$\mathrm{err}_{\mathrm{EB}} = \frac{2.58}{n}\sqrt{\frac{\mathring{\lambda}_1}{\lambda_1}\sum_{i=2}^{n}\frac{|\tilde{y}_i|^2}{\lambda_i}}, \qquad \theta_{\mathrm{EB}} = \operatorname*{argmin}_\theta\left[\log\left(\sum_{i=2}^{n}\frac{|\tilde{y}_i|^2}{\lambda_i}\right) + \frac{1}{n}\sum_{i=1}^{n}\log(\lambda_i)\right]$$

Example Integrands

$$\text{Gaussian probability} = \int_{[a,b]} \frac{\mathrm{e}^{-x^T \Sigma^{-1} x/2}}{(2\pi)^{d/2} |\Sigma|^{1/2}} \, \mathrm{d}x, \quad \text{(Genz, 1993)}$$



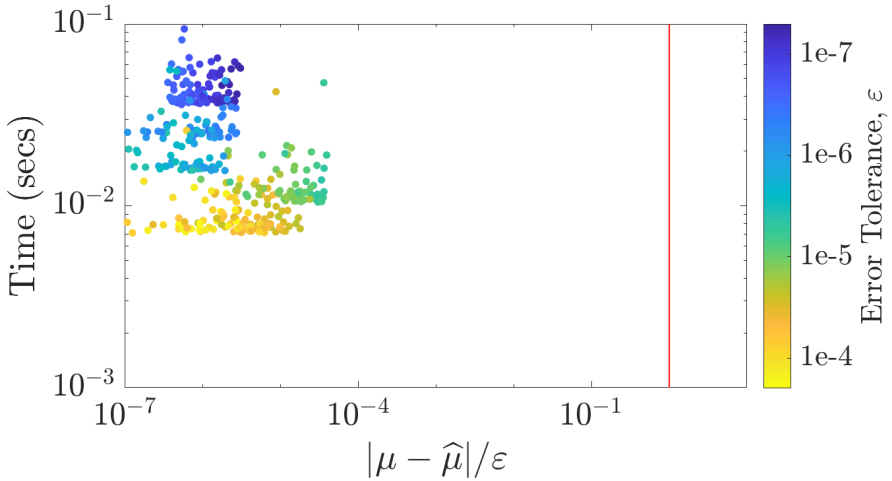
$$\text{Option pricing} = \int_{\mathbb{R}^d} \text{payoff}(x) \underbrace{\frac{\mathrm{e}^{-x^T \Sigma^{-1} x/2}}{(2\pi)^{d/2} |\Sigma|^{1/2}}}_{\text{PDF of Brownian motion at } d \text{ times}} \mathrm{d}x, \quad \text{(Glasserman, 2004)}$$

$$\text{where} \quad \text{payoff}(x) = \mathrm{e}^{-rT} \max\left(\frac{1}{d} \sum_{k=1}^{d} S_k(x_k) - K, 0 \right)$$

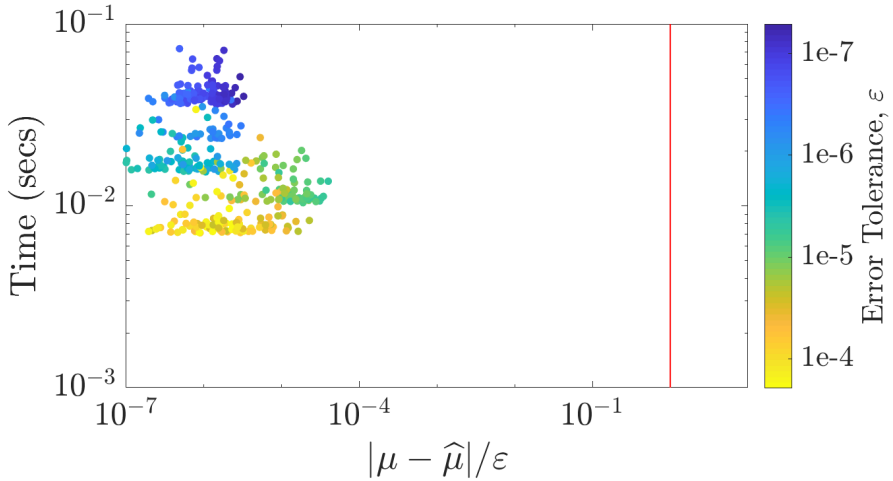
$$S_j(x_j) = S_0 \mathrm{e}^{(r-\sigma^2/2)t_j + \sigma x_j} = \text{stock price at time } t_j = jT/d;$$

$$\text{Keister integral} = \int_{\mathbb{R}^d} \cos(\|x\|) \exp(-\|x\|^2) \, \mathrm{d}x, \quad d = 1, 2, \ldots \quad \text{(Keister, 1996)}$$

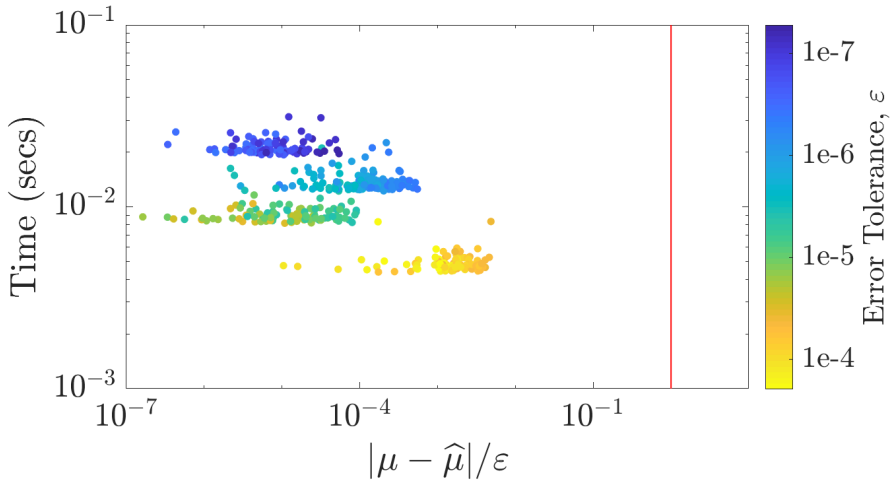
Multivariate normal probability: Lattice



(a) Empirical Bayes

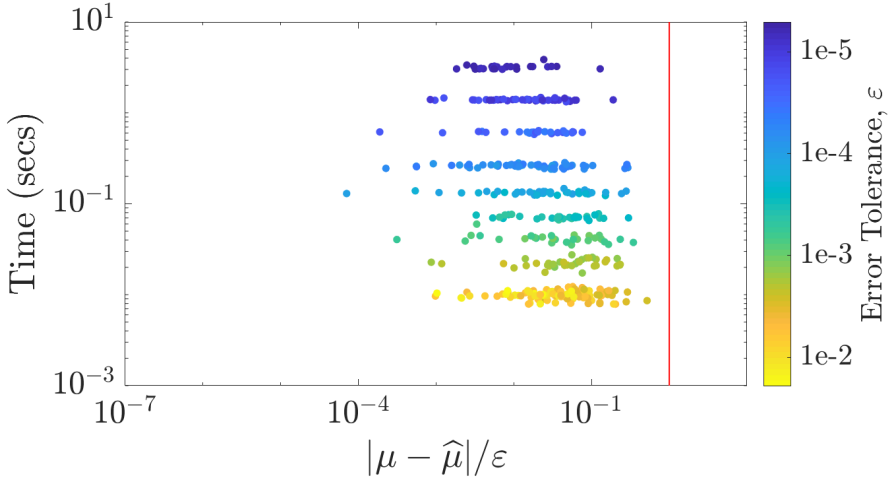


(b) Full Bayes

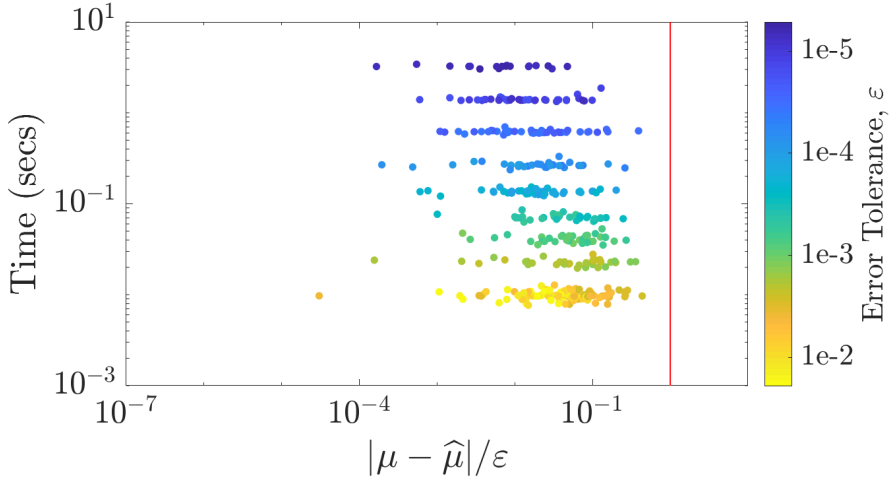


(c) GCV

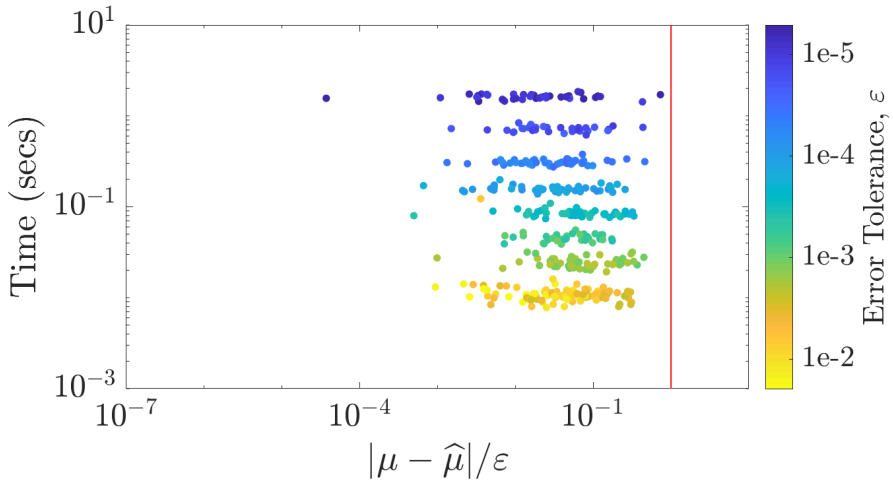
Multivariate normal probability: Sobol'



(a) Empirical Bayes

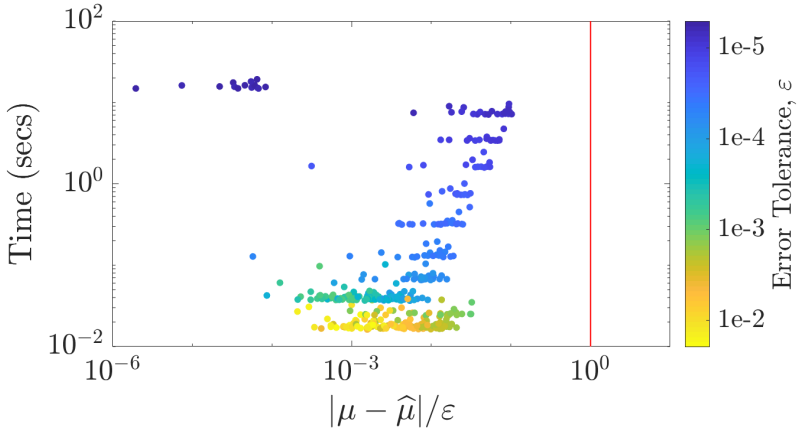


(b) Full Bayes

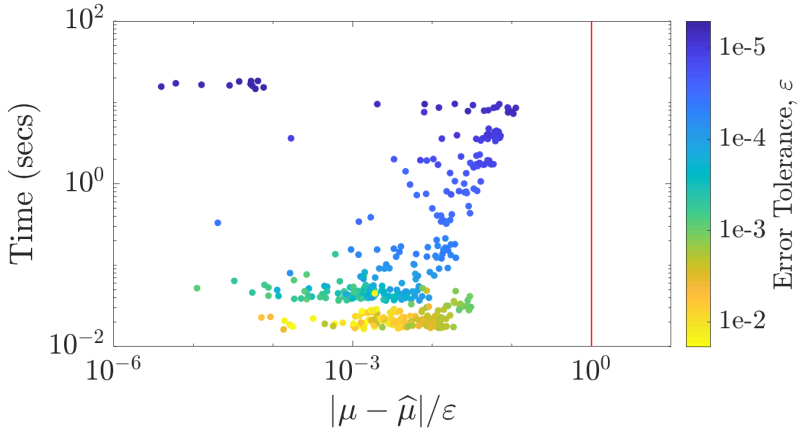


(c) GCV

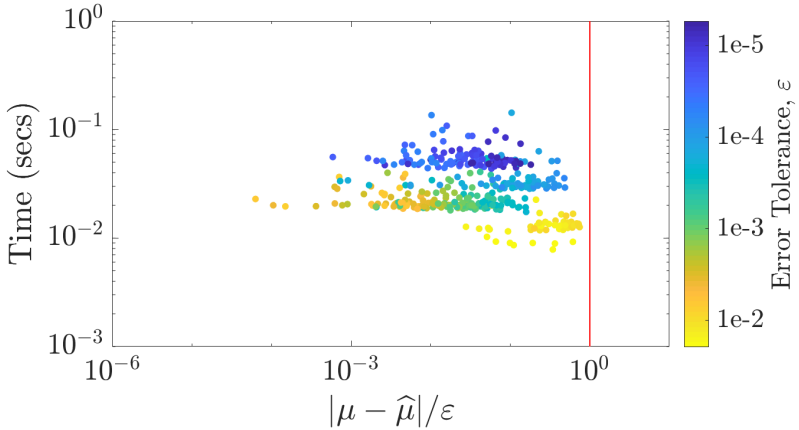
Keister Integral: Lattice



(a) Empirical Bayes

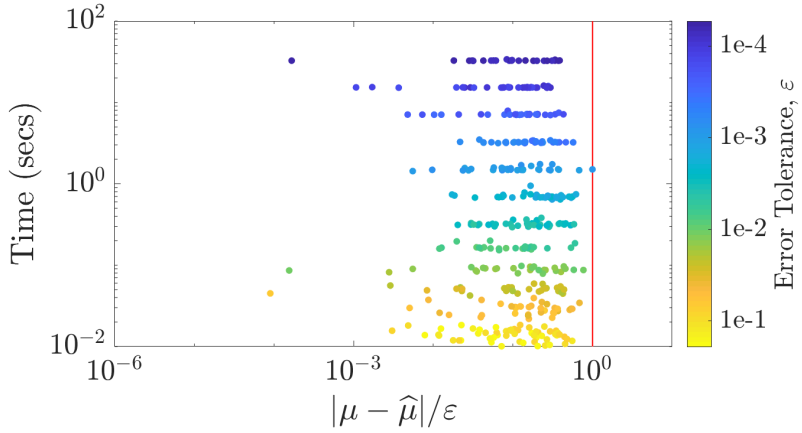


(b) Full Bayes

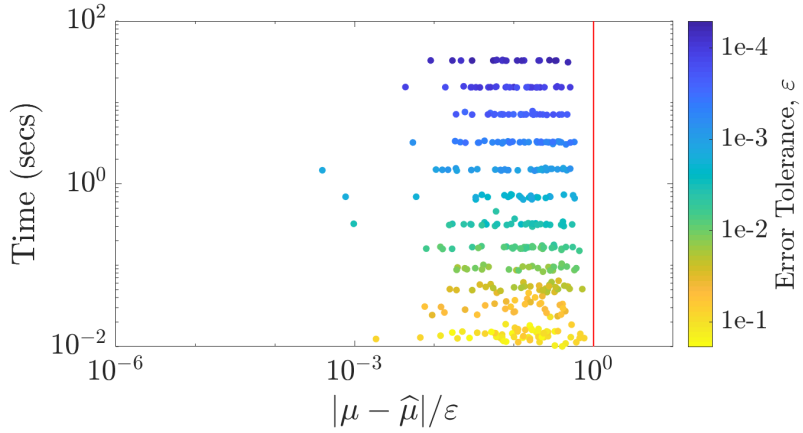


(c) GCV

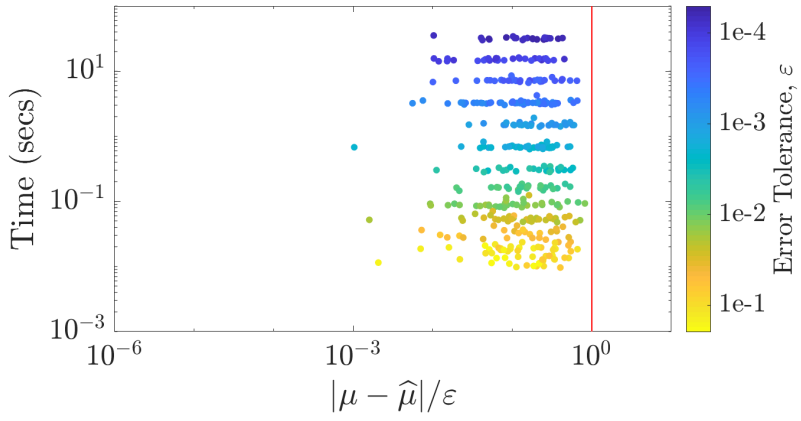
Keister Integral: Sobol'



(a) Empirical Bayes

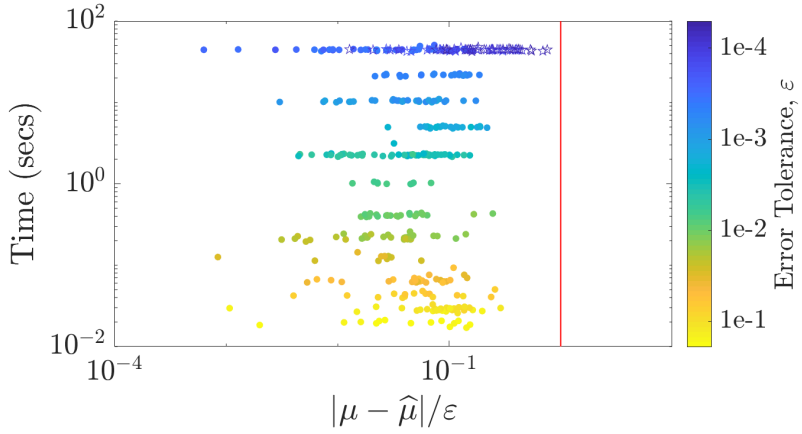


(b) Full Bayes

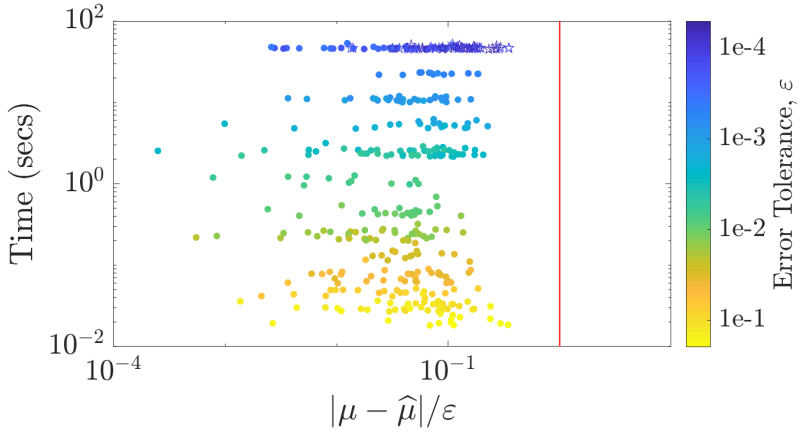


(c) GCV

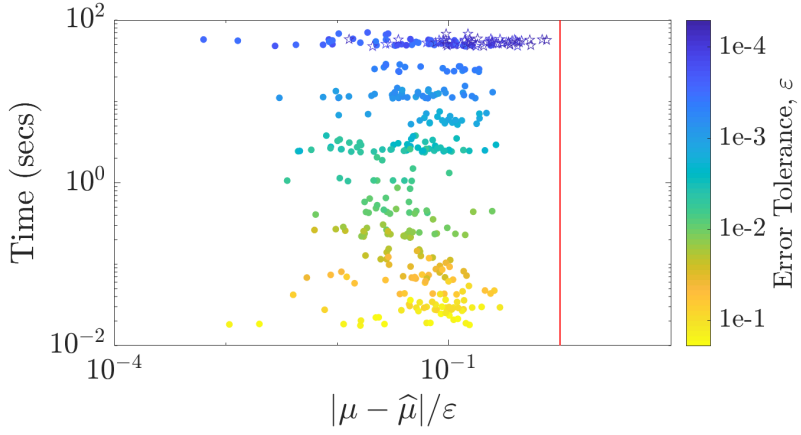
Option pricing: Lattice



(a) Empirical Bayes

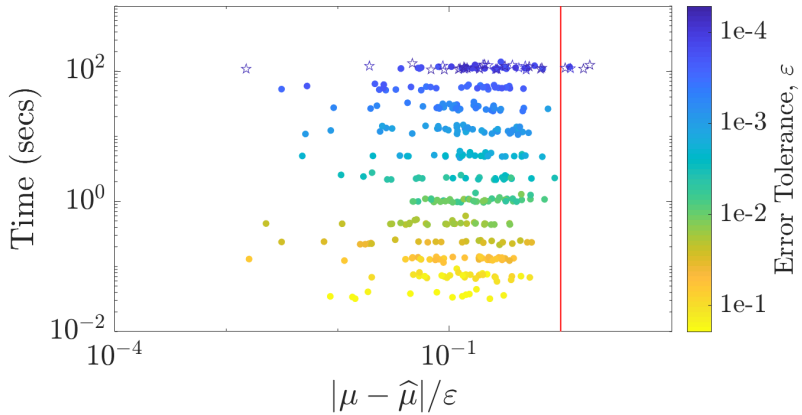


(b) Full Bayes

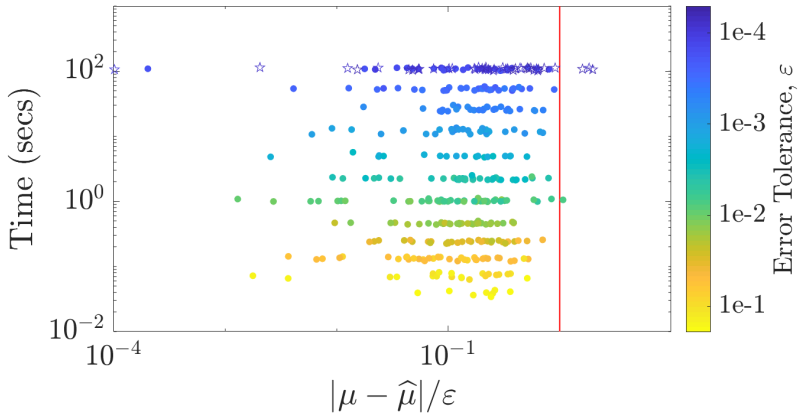


(c) GCV

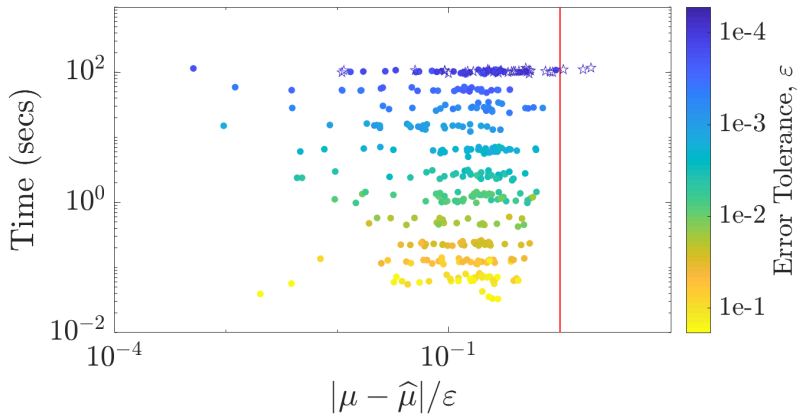
Option pricing: Sobol'



(a) Empirical Bayes



(b) Full Bayes



(c) GCV

Summary

- Developed a *technique* for a Fast Bayesian transform
- Developed two fast automatic Bayesian cubature algorithms with $\mathcal{O}(n \log n)$ complexity
- Having the advantages of a kernel method and the low computation cost of Quasi Monte carlo
- Scalable based on the complexity of the Integrand i.e, Kernel order and Lattice-points can be chosen to suit the smoothness of the integrand
- Conditioning problem if the kernel C is very smooth
- Source code : https://github.com/GailGithub/GAIL_Dev/tree/feature/BayesianCubature.
- A part of this work was published as a paper "Fast Automatic Bayesian Cubature Using Lattice Sampling by R. Jagadeeswaran, Fred J. Hickernell" (https://arxiv.org/abs/1809.09803)

Future work

- Choosing the kernel order r and periodization transform automatically
- Diagnostics for Gaussian process assumption
- Broaden the choice of numerical examples
- Better handling of conditioning problem and numerical errors
- Higher order nets and Walsh kernels could be used to achieve higher order or accuracy.

Future work : More applications

- Control variates : We would like to approximate a function of the form $(f - \beta_1 g_1 -, ..., -\beta_p g_p)$, then

$$f = \mathcal{N}\left(\beta_0 + \beta_1 g_1 +, ..., +\beta_p g_p, s^2 C_\theta\right)$$

- Function approximation : consider approximating a function of the form

$$\int_{[0,1]^d} \underbrace{f(\Phi(t)).\left|\frac{\partial \Phi}{\partial t}\right|}_{g(t)} \mathrm{d}t, \quad \text{where } \left|\frac{\partial \Phi}{\partial t}\right| \text{ is Jacobian, then}$$

$$g(\Psi(x)) = f(\underbrace{\Phi(\Psi(x))}_{x}).\left|\frac{\partial \Phi}{\partial t}\right|(\Psi(x)), \quad f(x) = g(\Psi(x)).\frac{1}{\left|\frac{\partial \Phi}{\partial t}\right|(\Psi(x))}$$

Finally, the function approximation is

$$\tilde{f}(x) = \tilde{g}(\Psi(x)) = \sum w_i C(.,.)$$

Thank you!

Parameter estimation - Full Bayes - General prior

What if $\rho_{m,s^2}(\xi,\lambda) \propto g(1/\lambda)$?

$$\rho_\mu(z|\boldsymbol{f}=\boldsymbol{y}) \propto \mathcal{LT}\{g(1/\cdot)\}^{\left(\frac{n-4}{2}\right)}(\chi)$$
$$\propto \mathcal{LT}\{g(1/\cdot)\}^{\left(\frac{n-4}{2}\right)}\left(1+\frac{(z-\hat{\mu}_{\text{full}})^2}{(n-1)\hat{\sigma}^2_{\text{full}}}\right)$$

Thus, $\rho_\mu(z|\boldsymbol{f}=\boldsymbol{y})$ is proportional to $\left(\frac{n-4}{2}\right)$th derivative of the Laplace transform of $g(1/\cdot)$ evaluated at χ, where $\chi \propto 1+\frac{(z-\hat{\mu}_{\text{full}})^2}{(n-1)\hat{\sigma}^2_{\text{full}}}$.

Our motivation to experiment with the general prior was to show that it may be possible to infer the prior from the integrand samples.

Periodization Transforms

Suppose the original integral is

$$\mu := \int_{(\boldsymbol{a},\boldsymbol{b})^d} g(\boldsymbol{t})\,\mathrm{d}\boldsymbol{t}, \quad \text{where } g \text{ is smooth, not periodic.}$$

The Baker's transform, the tent transform,

$$\boldsymbol{\Psi} : \boldsymbol{x} \mapsto (\Psi(x_1), \ldots, \Psi(x_d)), \quad \Psi(x) = 1 - 2\,|x - 1/2|\,, \quad f(\boldsymbol{x}) = g(\boldsymbol{\Psi}(\boldsymbol{x})).$$

A family of smoother variable transforms:

$$\boldsymbol{\Psi} : \boldsymbol{x} \mapsto (\Psi(x_1), \ldots, \Psi(x_d)), \quad \Psi : [0,1] \mapsto [0,1], \quad f(\boldsymbol{x}) = g(\boldsymbol{\Psi}(\boldsymbol{x})) \prod_{\ell=1}^{d} \Psi'(x_\ell).$$

Example:

$$C^1 : \Psi(x) = x^3(10 - 15x + 6x^2), \Psi'(x) = 30x^2(1-x)^2,$$

$$\text{Sidi's } C^1 : \Psi(x) = x - \frac{\sin(2\pi x)}{2\pi}, \Psi'(x) = 1 - \cos(2\pi x),$$

when it holds $\Psi \in C^{r+1}[0,1]$, $\lim_{x \downarrow 0} x^{-r-1}\Psi'(x) = \lim_{x \uparrow 1}(1-x)^{-r-1}\Psi'(x) = 0$, and $g \in C^{(r,\ldots,r)}[0,1]^d$, for $r \in \mathbb{N}_0$.

Walsh Transform

The WHT involves multiplications by $2^m \times 2^m$ Walsh-Hadamard matrices, which is constructed recursively, starting with $\mathsf{H}^{(0)} = 1$,

$$\mathsf{H}^{(1)} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

$$\mathsf{H}^{(2)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix},$$

$$\vdots$$

$$\mathsf{H}^{(m)} = \begin{pmatrix} \mathsf{H}^{(m-1)} & \mathsf{H}^{(m-1)} \\ \mathsf{H}^{(m-1)} & -\mathsf{H}^{(m-1)} \end{pmatrix} = \underbrace{\mathsf{H}^{(1)} \bigotimes \cdots \bigotimes \mathsf{H}^{(1)}}_{m \text{ times}} = \mathsf{H}^{(1)} \bigotimes \mathsf{H}^{(m-1)} \tag{12}$$

where $\bigotimes$ is Kronecker product.

Sobol' Nets and Walsh Kernels

Theorem

Any symmetric, positive definite, digital shift-invariant covariance kernel of the form (9) *scaled to satisfy* (4)*, when matched with digital net data-sites, satisfies assumptions* (3)*. The fast Walsh-Hadamard transform (FWHT) can be used to expedite the estimates of* θ *in* (5) *and the credible interval widths* (6) *in* $\mathcal{O}(n \log n)$ *operations. The cubature,* $\hat{\mu}$*, is just the sample mean.*

Walsh kernels + digital nets = 2×2 *block-Toeplitz* matrix

Eigenvectors of C_θ

The columns of Walsh-Hadamard matrix are the eigenvectors of C_θ, i.e., $V := H$

Theorem

Let $(x_i)_{i=0}^{n-1}$ be digitally shifted Sobol' nodes and K be any function, then the Gram matrix,

$$C_\theta = \big(C(x_i, x_j)\big)_{i,j=0}^{n-1} = \big(K(x_i \ominus x_j)\big)_{i,j=0}^{n-1},$$

where $n = 2^m$, $C(x, t) = K(x \ominus t)$, $x, t \in [0,1)^d$, is a 2×2 block-Toeplitz matrix and all the sub-blocks and their sub-sub-blocks, etc. are also 2×2 block-Toeplitz.

Iterative Computation of Walsh Transform

Let $\tilde{\boldsymbol{y}} = \mathsf{H}^{(m+1)}\boldsymbol{y}$ for some arbitrary $\boldsymbol{y} \in \mathbb{R}^{2n}$, $n = 2^m$. Define,

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{2n} \end{pmatrix}, \quad \boldsymbol{y}^{(1)} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \boldsymbol{y}^{(2)} = \begin{pmatrix} y_{n+1} \\ \vdots \\ y_{2n} \end{pmatrix},$$

$$\tilde{\boldsymbol{y}}^{(1)} = \mathsf{H}^{(m)}\boldsymbol{y}^{(1)} = \begin{pmatrix} \tilde{y}_1^{(1)} \\ \tilde{y}_2^{(1)} \\ \vdots \\ \tilde{y}_n^{(1)} \end{pmatrix}, \quad \tilde{\boldsymbol{y}}^{(2)} = \mathsf{H}^{(m)}\boldsymbol{y}^{(2)} = \begin{pmatrix} \tilde{y}_1^{(2)} \\ \tilde{y}_2^{(2)} \\ \vdots \\ \tilde{y}_n^{(2)} \end{pmatrix}.$$

Then,

$$\tilde{\boldsymbol{y}} = \mathsf{H}^{(m+1)}\boldsymbol{y} = \begin{pmatrix} \mathsf{H}^{(m)} & \mathsf{H}^{(m)} \\ \mathsf{H}^{(m)} & -\mathsf{H}^{(m)} \end{pmatrix} \begin{pmatrix} \boldsymbol{y}^{(1)} \\ \boldsymbol{y}^{(2)} \end{pmatrix}, \qquad \text{by (12)}$$

$$= \left(\mathsf{H}^{(m)}\boldsymbol{y}^{(1)} + \mathsf{H}^{(m)}\boldsymbol{y}^{(2)}\mathsf{H}^{(m)}\boldsymbol{y}^{(1)} - \mathsf{H}^{(m)}\boldsymbol{y}^{(2)}\right) = \begin{pmatrix} \tilde{\boldsymbol{y}}^{(1)} + \tilde{\boldsymbol{y}}^{(2)} \\ \tilde{\boldsymbol{y}}^{(1)} - \tilde{\boldsymbol{y}}^{(2)} \end{pmatrix} =: \tilde{\boldsymbol{y}}$$

Cone of functions and the Credible interval

In this research we assume that the integrand belongs to a cone of well-behaved functions, $\mathscr{C}$. Suppose that

$$|\mu(f) - \hat{\mu}_n(f)| \leqslant \text{err}_{\text{CI}}(f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_n)) \tag{13}$$

for some f, which it is 99% of the time under our hypothesis. Also note that our $\text{err}_{\text{CI}}, \text{CI} \in \{\text{EB}, \text{GCV}\}$ are positively homogeneous functions, meaning,

$$\text{err}_{\text{CI}}(ay_1, \cdots, ay_n) = |a| \, \text{err}_{\text{CI}}(y_1, \cdots, y_n).$$

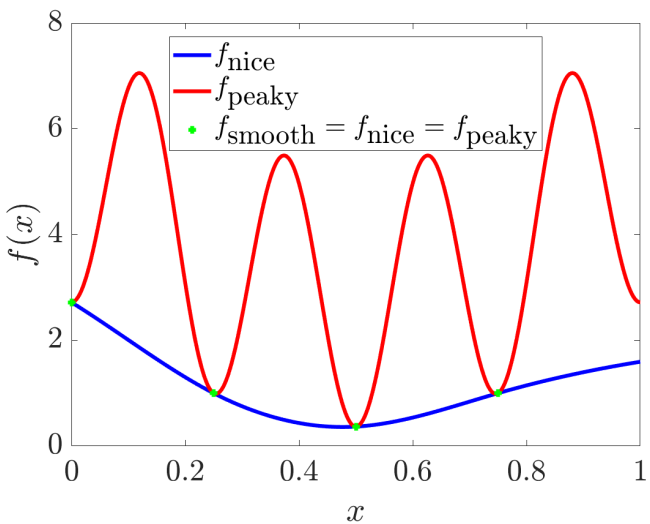
Thus if f satisfies (13), then

$$\begin{aligned} |\mu(af) - \hat{\mu}_n(af)| &= |a| \, |\mu(f) - \hat{\mu}_n(f)| \\ &\leqslant |a| \, \text{err}_{\text{CI}}(f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_n)) = \text{err}_{\text{CI}}(af(\boldsymbol{x}_1), \cdots, af(\boldsymbol{x}_n)) \end{aligned}$$

for all real a. Thus the set of all f satisfying (13) is a *cone*, $\mathscr{C}$. Cones of functions satisfy the property that if $f \in \mathscr{C}$ then $af \in \mathscr{C}$.

$$\mathbb{P}_f \left[|\mu(f) - \hat{\mu}_n(f)| \leqslant \text{err}_{\text{CI}}(f) \right] \geqslant 99\%.$$

Let $f_{\text{TRUE}}(\boldsymbol{x}) = \exp(\sum_{\ell=1}^{d} \cos(2\pi x_\ell))$ and the peaky integrand $f_{\text{PEAKY}}(\boldsymbol{x}) = f_{\text{TRUE}} + a_{\text{PEAKY}} f_{\text{NOISE}}$, $f_{\text{NOISE}}(\boldsymbol{x}) = (1 - \exp(2\pi\sqrt{-1}\boldsymbol{x}^T\boldsymbol{\zeta}))$, $\boldsymbol{\zeta} \in \mathbb{R}^d$ is some d-dimensional vector belonging to the dual space of the lattice nodes. The f_{NICE} is obtained by kernel interpolation.



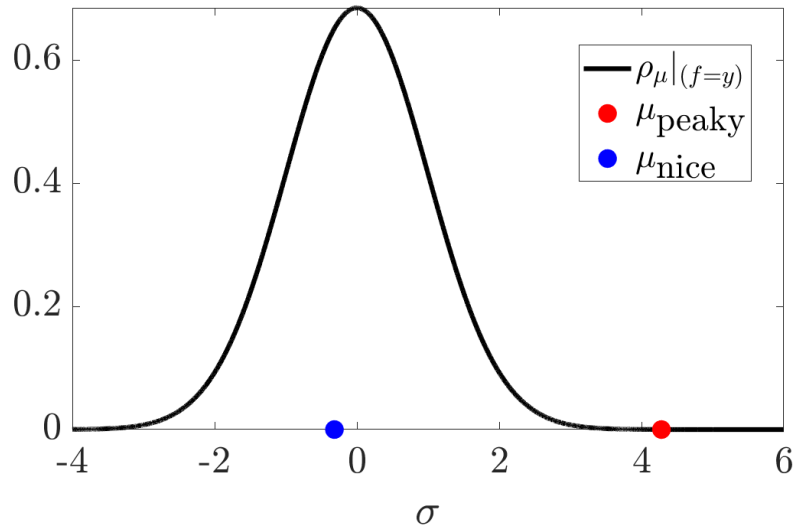


Figure: Left: Example integrands 1) f_{NICE}, a smooth function, 2) f_{PEAKY}, a peaky function. The function values $f_{\text{PEAKY}}(\boldsymbol{x}_i) = f_{\text{NICE}}(\boldsymbol{x}_i) = f_{\text{TRUE}}(\boldsymbol{x}_i)$ for $i = 1, \cdots, n$. Right: Probability distributions showing the relative integral position of a smooth and a peaky function. f_{NICE} lies within the center 99% of the confidence interval, and f_{PEAKY} lies on the outside of 99% of the confidence interval.

Shape parameter search using gradient descent

Steepest descent search is defined as:

$$\eta_\ell^{(j+1)} = \eta_\ell^{(j)} - \nu \frac{\partial}{\partial \eta_\ell} \mathcal{L}_{\mathsf{x}}(\theta|\boldsymbol{y}), \quad j = 0, 1, \cdots, \quad \ell = 1, \cdots, d$$

$$\mathsf{x} \in \{\mathsf{EB}, \mathsf{GCV}\}$$

where ν is the step size for the gradient descent, j is the iteration index, and $\frac{\partial}{\partial \eta_\ell} \mathcal{L}(\theta|\boldsymbol{y})$ is either (14) or (15) depending on the choice of the hyperparameter search method. The parameter η_ℓ is usually searched in the whole $\mathbb{R}$ by using the simple domain transformation.

Computing the derivative of $\mathcal{L}_{\text{EB}}(\theta|\boldsymbol{y})$

Taking derivative with respect to θ_ℓ, for $\ell = 1, \cdots, d$

$$\mathcal{L}_{\text{EB}}(\theta|\boldsymbol{y}) = \log\left((\boldsymbol{y} - m_{\text{EB}}\mathbf{1})^T \mathsf{C}_\theta^{-1}(\boldsymbol{y} - m_{\text{EB}}\mathbf{1})\right) + \frac{1}{n}\log(\det(\mathsf{C}_\theta)),$$

$$\frac{\partial}{\partial\theta_\ell}\mathcal{L}_{\text{EB}}(\theta|\boldsymbol{y}) = -\frac{\left((\boldsymbol{y} - m_{\text{EB}}\mathbf{1})^T \mathsf{C}_\theta^{-1}\right)^T \left(\frac{\partial \mathsf{C}}{\partial\theta_\ell}\right)\left((\boldsymbol{y} - m_{\text{EB}}\mathbf{1})^T \mathsf{C}_\theta^{-1}\right)}{(\boldsymbol{y} - m_{\text{EB}}\mathbf{1})^T \mathsf{C}_\theta^{-1}(\boldsymbol{y} - m_{\text{EB}}\mathbf{1})}$$

$$+ \frac{1}{n}\text{trace}\left(\mathsf{C}_\theta^{-1}\frac{\partial \mathsf{C}}{\partial\theta_\ell}\right), \quad \text{where} \quad m_{\text{EB}} = \frac{\mathbf{1}^T \mathsf{C}_\theta^{-1}\boldsymbol{y}}{\mathbf{1}^T \mathsf{C}_\theta^{-1}\mathbf{1}},$$

where we used some of the results from (Dong et al., 2017). After using the fast Bayesian transform properties

$$\frac{\partial}{\partial\theta_\ell}\mathcal{L}_{\text{EB}}(\theta|\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n}\frac{\bar{\lambda}_{i(\ell)}}{\lambda_i} - \left(\sum_{i=2}^{n}\frac{|\tilde{\boldsymbol{y}}_i|^2\,\bar{\lambda}_{i(\ell)}}{\lambda_i^2}\right)\left(\sum_{i=2}^{n}\frac{|\tilde{\boldsymbol{y}}_i|^2}{\lambda_\ell}\right)^{-1} \tag{14}$$

Computing the derivative of $\mathcal{L}_{\text{GCV}}(\theta|\boldsymbol{y})$

Similarly for the generalized cross-validation

$$\mathcal{L}_{\text{GCV}}(\theta|\boldsymbol{y}) = \log\left(\boldsymbol{y}^T\left[\mathsf{C}_\theta^{-2} - \frac{\mathsf{C}_\theta^{-2}\mathbf{1}\mathbf{1}^T\mathsf{C}_\theta^{-2}}{\mathbf{1}^T\mathsf{C}_\theta^{-2}\mathbf{1}}\right]\boldsymbol{y}\right) - \log\left(\text{trace}(\mathsf{C}_\theta^{-2})\right),$$

$$\text{where} \quad m_{\text{GCV}} = \frac{\mathbf{1}^T\mathsf{C}_\theta^{-2}\boldsymbol{y}}{\mathbf{1}^T\mathsf{C}_\theta^{-2}\mathbf{1}},$$

After using the fast Bayesian transform properties

$$\frac{\partial}{\partial\theta_\ell}\mathcal{L}_{\text{GCV}}(\theta|\boldsymbol{y}) = -2\left(\sum_{i=2}^{n}\frac{|\widetilde{y}_i|^2}{\lambda_i^2}\right)^{-1}\left(\sum_{i=2}^{n}\frac{|\widetilde{y}_i|^2\,\bar{\lambda}_{i(\ell)}}{\lambda_i^3}\right) + 2\left(\sum_{i=1}^{n}\frac{1}{\lambda_i}\right)^{-1}\left(\sum_{i=1}^{n}\frac{\bar{\lambda}_{i(\ell)}}{\lambda_i^2}\right), \tag{15}$$

where $\bar{\lambda}_{i(\ell)}$ is the derivative of the ith eigenvalue of the Gram matrix, C, in the ℓth variable.

Product Kernels

Product kernels in d dimensions are of the form,

$$C_\theta(\boldsymbol{t},\boldsymbol{x}) = \prod_{\ell=1}^{d}\Big[1-\eta_\ell\ \mathfrak{C}(x_\ell,t_\ell)\Big] \tag{16}$$

where η_ℓ is called shape parameter.

Derivative of the product kernel when $\eta_1 = \cdots = \eta_d = \eta$

$$\frac{\partial}{\partial\eta}C_\theta(\boldsymbol{t},\boldsymbol{x}) = (d/\eta)C_\theta(\boldsymbol{t},\boldsymbol{x})\left(1-\frac{1}{d}\sum_{\ell=1}^{d}\frac{1}{1-\eta\mathfrak{C}(x_\ell,t_\ell)}\right).$$

When η_ℓ is different for each $\ell = 1,\cdots,d$

$$\frac{\partial}{\partial\eta_\ell}C_\theta(\boldsymbol{t},\boldsymbol{x}) = \frac{1}{\eta_\ell}C_\theta(\boldsymbol{t},\boldsymbol{x})\left(1-\frac{1}{1-\eta_\ell\mathfrak{C}(x_\ell,t_\ell)}\right).$$

To compute $\bar{\lambda}_{i(\ell)}$

If V does not depend on θ then one can fast compute the derivative of Gram matrix C,

$$\frac{\partial \mathsf{C}}{\partial \theta_\ell} = \frac{1}{n}\mathsf{V}\frac{\partial \Lambda}{\partial \theta_\ell}\mathsf{V}^H = \frac{1}{n}\mathsf{V}\bar{\Lambda}_{(\ell)}\mathsf{V}^H, \quad \text{using} \quad \mathsf{C} = \frac{1}{n}\mathsf{V}\Lambda\mathsf{V}^H$$

$$\text{where} \quad \bar{\Lambda}_{(\ell)} = \mathrm{diag}(\bar{\boldsymbol{\lambda}}_{(\ell)}), \quad \text{and}$$

$$\bar{\boldsymbol{\lambda}}_{(\ell)} = \frac{\partial \boldsymbol{\lambda}}{\partial \theta_\ell} = \left(\frac{\partial \lambda_i}{\partial \theta_\ell}\right)_{i=1}^{n} = \left(\frac{\partial}{\partial \theta_\ell}\mathsf{V}^H \boldsymbol{C}_1\right) = \mathsf{V}^H \left(\frac{\partial}{\partial \theta_\ell} C_\theta(\boldsymbol{x}_1, \boldsymbol{x}_i)\right)_{i=1}^{n}, \tag{17}$$

where we used the fast Bayesian transform property $\boldsymbol{\lambda} = \mathsf{V}^H C_1$.

References I

Briol, F-X, C. J. Oates, M. Griolami, M. A. Osborne, and D. Sejdinovic. 2018+. *Probabilistic integration: A role in statistical computation?*, Statist. Sci. to appear.

Diaconis, P. 1988. *Bayesian numerical analysis*, Statistical decision theory and related topics iv, papers from the 4th purdue symp., west lafayette/indiana 1986, pp. 163–175.

Dick, J. and F. Pillichshammer. 2010. *Digital nets and sequences: Discrepancy theory and quasi-Monte Carlo integration*, Cambridge University Press, Cambridge.

Dong, K., D. Eriksson, H. Nickisch, D. Bindel, and A. G. Wilson. 2017. *Scalable log determinants for gaussian process kernel learning*, NIPS. in press.

Genz, A. 1993. *Comparison of methods for the computation of multivariate normal probabilities*, Computing Science and Statistics **25**, 400–405.

Glasserman, P. 2004. *Monte Carlo methods in financial engineering*, Applications of Mathematics, vol. 53, Springer-Verlag, New York.

Hickernell, F. J. and H. Niederreiter. 2003. *The existence of good extensible rank-1 lattices*, J. Complexity **19**, 286–300.

Keister, B. D. 1996. *Multidimensional quadrature algorithms*, Computers in Physics **10**, 119–122.

Nuyens, Dirk. 201308. *The construction of good lattice rules and polynomial lattice rules*.

References II

O'Hagan, A. 1991. *Bayes-Hermite quadrature*, J. Statist. Plann. Inference **29**, 245–260.

Olver, F. W. J., D. W. Lozier, R. F. Boisvert, C. W. Clark, and A. B. O. Dalhuis. 2013. *Digital library of mathematical functions*.

Rasmussen, C. E. and C. Williams. 2003. *Bayesian Monte Carlo*, Advances in Neural Information Processing Systems, pp. 489–496.

Ritter, K. 2000. *Average-case analysis of numerical problems*, Lecture Notes in Mathematics, vol. 1733, Springer-Verlag, Berlin.

Sobol', I. M. 1976. *Uniformly distributed sequences with an additional uniformity property*, Zh. Vychisl. Mat. i Mat. Fiz. **16**, 1332–1337 (Russian).

Traub, J. F., G. W. Wasilkowski, and H. Woźniakowski. 1988. *Information-based complexity*, Academic Press, Boston.