Markov Chain Cubature for Bayesian Inference

James Foster
University of Bath

joint with Daniel Burrows (Bath), Terry Lyons (Oxford), Harald Oberhauser (Oxford) and Tom L (GCHQ)

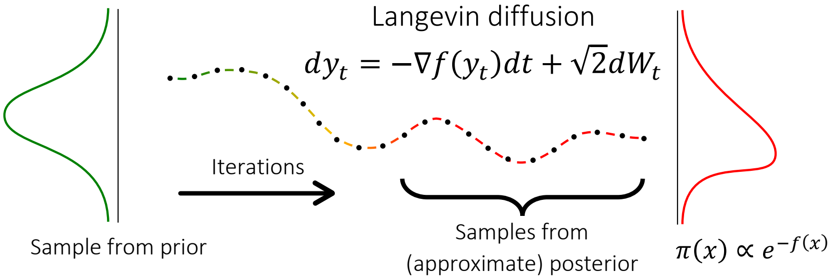
DataSıg

A rough path between mathematics and data science

Outline

Markov Chain Monte Carlo

A fundamental challenge in Bayesian inference is computing integrals with respect to posterior distributions and Markov Chain Monte Carlo (MCMC) is widely regarded as the "go-to" approach for these problems.



Under mild conditions on $f : \mathbb{R}^d \to \mathbb{R}$, the Langevin SDE admits a unique strong solution that is ergodic with stationary measure $\pi(x) \propto e^{-f(x)}$ [1].

Unadjusted Langevin Algorithm (ULA): $\boxed{Y_{n+1} := Y_n - \nabla f(Y_n)h + \sqrt{2}W_n}$

Convergence of Langevin Monte Carlo

For sufficiently small h, ULA "ends up close" to the target distribution [2]. However, ULA is a Markov chain and might not explore the space quickly.

Broadly speaking, this leads to two possible approaches

- Run N independent ULA chains and sample points at a fixed time T.
- Run a ULA chain over a long time horizon and sample at each step.

The latter strategy is much preferred by practitioners, but relies on the Markov chain to have a **fast mixing time**.

Motivating questions

Can Langevin dynamics be simulated as a cloud of (dependent) points?

Accuracy? Computational cost? Exploration of parameter space?

Outline

One-step cubature formula

Recall the Langevin diffusion with invariant measure $\pi \propto e^{-f}$ is given by

$$dy_t = -\nabla f(y_t)\,dt + \sqrt{2}\,dW_t, \tag{1}$$

By Taylor expanding (1), we can see that y has the following moments:

$$\mathbb{E}\big[(y_t - y_s)\,|\,y_s\big] = -\nabla f(y_s)\,h + O(h^2), \tag{2}$$

$$\mathbb{E}\big[(y_t - y_s)^{\otimes 2}\,|\,y_s\big] = 2hI_d + O(h^2), \tag{3}$$

for $s, t \geq 0$, where $h = t - s > 0$.

Construction of a first order one-step cubature rule started at y_s

We want to construct a cloud of points and weights $\{(x_i, w_i)\}_{1\leq i\leq N}$ with

$$\mu := \sum_{i=1}^{N} w_i x_i = -\nabla f(y_s)\,h, \qquad \Sigma := \sum_{i=1}^{N} w_i (x_i - \mu)(x_i - \mu)^{\top} = 2hI_d.$$

One-step cubature formula via Hadamard matrices

The Hadamard (or Walsh) matrices are defined inductively as

$$H_1 := (1)\,, \quad H_{2^{k+1}} := \begin{pmatrix} H_{2^k} & H_{2^k} \\ H_{2^k} & -H_{2^k} \end{pmatrix}.$$

Let $n := 2^{\lceil \log d \rceil}$ and define n vectors $\{e_i\}_{1 \leq i \leq n}$ in $\mathbb{R}^d$ as columns of H_n

$$H_n = \begin{pmatrix} e_1 & e_2 & \cdots & e_{n-1} & e_n \\ & & (n-d) \times n \ \text{matrix} & & \end{pmatrix},$$

and another vectors $\{e_i\}_{n+1 \leq i \leq 2n}$ as $e_i := -e_{i-n}$ for $n+1 \leq i \leq 2n$.

Theorem (Victoir (2005))

Let $X \sim \mathit{Uniform}\big(\{e_i\}_{1 \leq i \leq 2n}\big)$. Then

$$\mathbb{E}\big[X\big] = 0, \ \ \mathbb{E}\big[X^{\otimes 2}\big] = I_d\,, \ \ \mathbb{E}\big[X^{\otimes 3}\big] = 0.$$

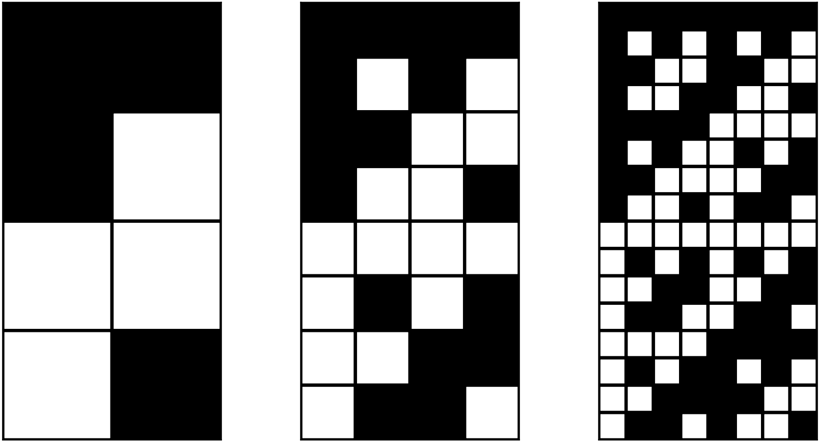
Thus, X matches the first 3 moments of the standard normal distribution.

One-step cubature formula via Hadamard matrices

H_2

H_4

H_8

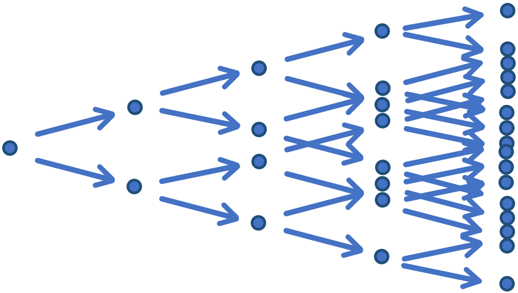


One-step cubature formula via Hadamard matrices

Therefore, in each step, we can define $2^{\lceil \log d \rceil + 1}$ new cubature points as

$$y_i^{\text{new}} := y_s - \nabla f(y_s)h + \sqrt{2h}e_i, \tag{4}$$

each with equal weight. The moments of (4) will then have $O(h^2)$ error. Equation (4) is a particular example of "Cubature on Weiner Space" [4]. However, even in the one-dimensional setting, there is a clear problem!



After each step, the number of points increases by a factor of $\approx 2d$.

Outline

Distribution compression

Thus, for SDE cubature to be practical, we require an algorithm that can "compress" (or reduce the support of) discrete probability distributions. Whilst we shall just resample points, there are sophisticated algorithms where the compressed measure accurately integrates certain functions.

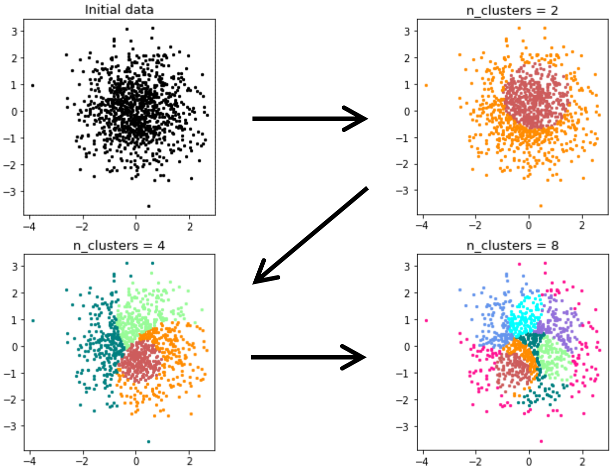
Test functions are explicitly specified by the user (e.g. polynomials)

- High order Recombination (Litterer and Lyons (2012), Tchernychova and Lyons (2016), Cosentino et al. (2020))

Test functions come from a reproducing kernel (i.e. MMD distance)

- Kernel Herding (Chen et al. (2010))
- Kernel Thinning (Dwivedi and Mackey, (2021))
- Kernel Recombination (Hayakawa et al. (2022))
- Stein Thinning (Riabiz et al. (2022))

Applying distribution compression locally (ball tree)



A standard ball tree algorithm for partitioning points is implemented in

from sklearn.neighbors import BallTree

A basic cubature algorithm

Step 0. Generate a cloud of N points Y_0 with uniform weights w_0 from a prior distribution.

Step 1. In each step, generate a new cloud of $2^{1+\lceil \log d \rceil} N$ points by

$$Y_{n+1,i} := Y_n - \nabla f(Y_n)h + \sqrt{2h}e_i,$$

with weights $w_{n+1,i} := 2^{-(1+\lceil \log d \rceil)} w_n$.

The vectors $\{e_i\}$ come from the “Hadamard” cubature formula.

Step 2. Put the points into N smaller “patches” via a ball tree algorithm.

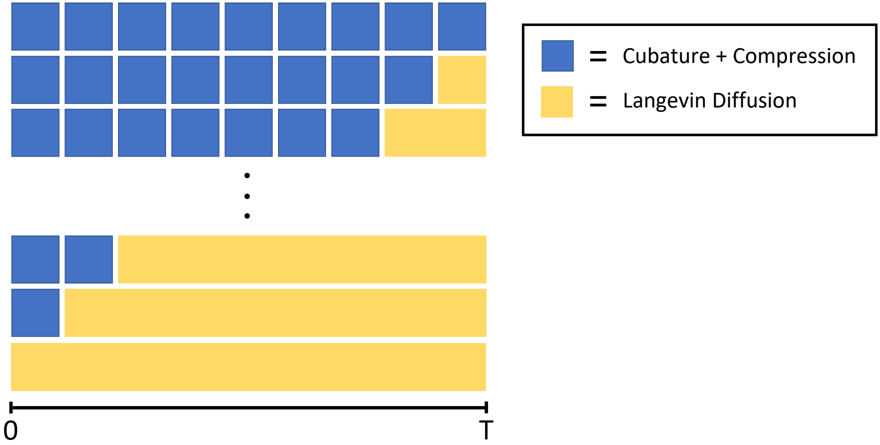
Step 3. On each patch, resample a point (or reduce small clouds of points/weights using a distribution compression algorithm).

Step 4. Repeat steps 1 – 3.

Outline

Error analysis of cubature

We want to compare N steps of Langevin cubature to the SDE at $T = Nh$.



Error analysis of cubature

For sufficiently smooth test functions $F : \mathbb{R}^d \to \mathbb{R}$, let

$$P_t F : y \mapsto \mathbb{E}\big[F(y_t) \,|\, y_0 = y \big],$$
$$Q_k F : Y \mapsto \mathbb{E}\big[F(Y_k) \,|\, Y_0 = Y \big],$$

denote the semigroups corresponding to the Langevin diffusion and the "one-step cubature + compression" step. By the telescoping sum trick,

$$\begin{aligned} P_T F - Q_N F &= \sum_{k=0}^{N-1} Q_{N-(k+1)}\big(P_{t_{k+1}} F\big) - Q_{N-k}\big(P_{t_k} F\big) \\ &= \sum_{k=0}^{N-1} Q_{N-(k+1)}\big(P_h - Q_1\big)\big(P_{t_k} F\big), \end{aligned}$$

where the cubature algorithm uses a step size of $h > 0$ and $t_k := kh$.

Error analysis of cubature

Since Q_1 is a Markov operator, it follows from [12, Theorem 13.2] that

$$\begin{aligned}\big\|P_T F - Q_N F\big\|_\infty &\leq \sum_{k=0}^{N-1}\big\|Q_{N-(k+1)}\big(P_h - Q_1\big)\big(P_{t_k} F\big)\big\|_\infty\\ &= \sum_{k=0}^{N-1}\Big\|\underbrace{Q_1\cdots Q_1}_{\substack{N-(k+1)\\ \text{times}}}\big(P_h - Q_1\big)\big(P_{t_k} F\big)\Big\|_\infty\\ &\leq \sum_{k=0}^{N-1}\big\|\big(P_h - Q_1\big)\big(P_{t_k} F\big)\big\|_\infty.\end{aligned}$$

Hence, there are two different operators to estimate: $(P_h - Q_1)$ and P_t.

Error analysis of cubature (based on Talay-Tubaro [13])

The first term, $P_h F$, can be understood "simply" by using Itô's lemma.

$$\mathbb{E}\big[F(y_h)\big] = F(y_0) + \big(\Delta F(y_0) - \nabla F(y_0)\nabla f(y_0)\big)h + R_1(y_0)h^2,$$

where

- $F \in \mathcal{C}^4(\mathbb{R}^d)$ with F and its first 3 derivatives Lipscthiz continuous
- $f \in \mathcal{C}^3(\mathbb{R}^d)$ with f and its first 2 derivatives Lipschitz continuous
- The remainder $R_1(y_0)$ satisfies a "linear in F" upper bound

$$\begin{aligned}\|R_1\|_\infty \leq \frac{1}{2}\Big(&\|\Delta^2 F\|_\infty + 2\|\Delta(\nabla F)\|_\infty\|\nabla f\|_\infty + \|\nabla^2 F\|_\infty\|\nabla f\|_\infty^2 \\ &+ \|\nabla F\|_\infty\|\Delta(\nabla f)\|_\infty + \|\nabla F\|_\infty\|\nabla^2 f\|_\infty\|\nabla f\|_\infty\Big).\end{aligned}$$

Error analysis of cubature (based on Talay-Tubaro [13])

The second term, Q_1F, can be estimated “simply” by Taylor expansion.

$$\mathbb{E}\Big[F\big(y - \nabla f(y)h + \sqrt{2h}\,Z\big)\Big] = F(y) + \big(\Delta F(y) - \nabla F(y)\nabla f(y)\big)h + R_2(y)h^2,$$

where

- $Z \sim \text{Uniform}\big(\{e_i\}_{1\leq i\leq 2n}\big)$
- $F \in \mathcal{C}^4(\mathbb{R}^d)$ with F and its first 3 derivatives Lipscthiz continuous
- $f \in \mathcal{C}^3(\mathbb{R}^d)$ with f and its first 2 derivatives Lipschitz continuous
- The remainder $R_2(y)$ satisfies a “linear in F” upper bound

$$\begin{aligned}\|R_2\|_\infty \leq \frac{1}{2}\|\nabla^2 F\|_\infty\|\nabla f\|_\infty^2 + \|\Delta(\nabla F)\|_\infty\|\nabla f\|_\infty + \frac{1}{6}\|\nabla^3 F\|_\infty\|\nabla f\|_\infty^3 h \\ + \frac{1}{24}\|\nabla^4 F\|_\infty\Big(\|\nabla f\|_\infty^4 h^2 + 4\sqrt{2}\,\|\nabla f\|_\infty^3 d h^{\frac{3}{2}} + 12\|\nabla f\|_\infty^2 d^2 h \\ + 8\sqrt{2}\,\|\nabla f\|_\infty d^3 h^{\frac{1}{2}} + 4d^4\Big).\end{aligned}$$

Error analysis of cubature (based on Talay-Tubaro [13])

Theorem (Smoothing of the semigroup (Leimkuhler et al., 2014))

Let $f \in \mathcal{C}^7(\mathbb{R}^d)$ have a Lipschitz gradient and higher derivatives bounded. Suppose $F \in \mathcal{C}^6(\mathbb{R}^d)$ is a function such that it and its derivatives grow no faster than a polynomial at infinity.

We assume a dissipativity condition, that $\exists c_0 \in \mathbb{R}$ and $c_1 > 0$ such that

$$\langle x, \nabla f(x)\rangle \geq c_0 + c_1\|x\|_2^2,$$

for all $x \in \mathbb{R}^d$. Then there exists $C_F > 0$, $K \in \mathbb{N}$ and $\lambda > 0$ such that

$$\Big|\big(P_t F\big)(x) - \big(P_\infty F\big)(x)\Big| \leq C_F\big(1 + \|x\|_2^K\big)e^{-\lambda t},$$

$$\left|\frac{\partial^{j+|\boldsymbol{i}|}}{\partial^j t\, \partial^{i_1} x^1 \cdots \partial^{i_d} x^d}\big(P_t F\big)(x)\right| \leq C_F\big(1 + \|x\|_2^K\big)e^{-\lambda t},$$

for all $1 \leq |\boldsymbol{i}| \leq 8$ and $0 \leq j \leq 2$.

Error analysis of cubature (based on Talay-Tubaro [13])

Putting this all together...

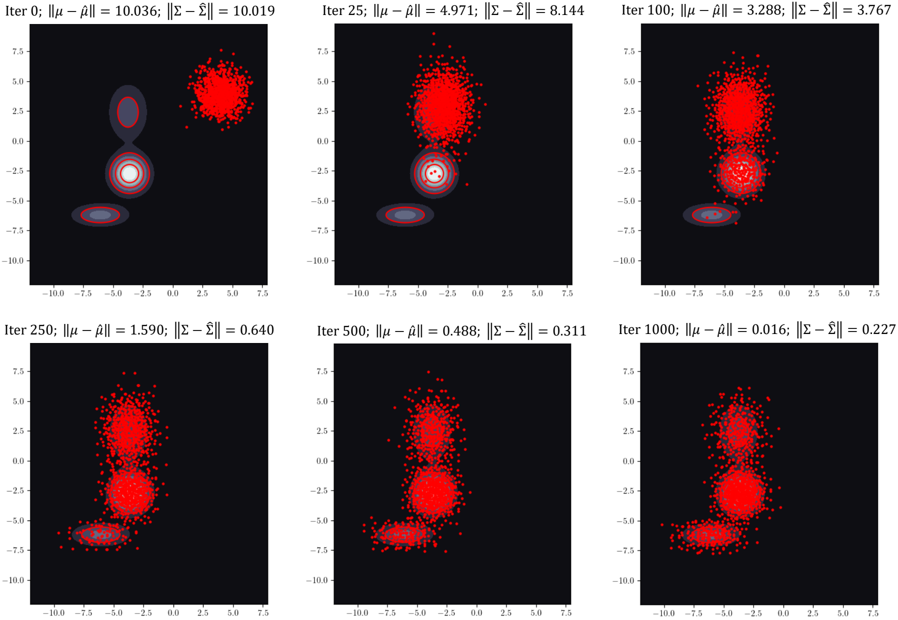
$$\begin{aligned}\big\|P_T F - Q_N F\big\|_\infty &\leq \sum_{k=0}^{N-1}\big\|\big(P_h - Q_1\big)\big(P_{t_k} F\big)\big\|_\infty \\ &\leq \sum_{k=0}^{N-1} Ch^2 e^{-\lambda t_k} \\ &\leq C\Big(\frac{1}{\lambda} + h\Big)h.\end{aligned}$$

Just like in the analysis of MCMC algorithms, this does not depend on N!

Provided each "cubature + compression" step integrates smooth functions as P_h with an $o(h)$ error, and keeps moments uniformly bounded, then the Langevin cubature can be applied over $[0, \infty)$.

Outline

Langevin cubature on a Gaussian mixture model



Cubature vs MCMC on a Gaussian mixture model

All algorithms were implemented in Python and ran on a laptop (not in parallel)

Algorithm	Time (s)	Iterations	$\|\mu - \hat{\mu}\|$ (3.d.p)
Langevin cubature	32.9	1000	0.016
Single MCMC chain (ULA)	362.5	1000000 (+ 1000 burn-in)	0.047

Cubature parameters:

- Step size, $h = 0.1$
- Number of particles, $N = 1024$

MCMC parameters:

- Step size, $h = 0.1$

Cubature vs SVGD on Bayesian logistic regression

Forest Covertype dataset [15]: 581,012 data points and 54 features.

Algorithm	Iterations	Test accuracy	Log-likelihood
Langevin Cubature (with Tamed Euler [16])	1500	75.9%	-0.564
Stein Variational Gradient Descent	6000	75.7%	-0.521

Langevin Cubature ran for 10.7 hours whereas SVGD ran for 7.1 hours. SVGD has similar test accuracy, but is much faster with fewer particles!

Cubature parameters:

- Step size, $h = 0.01$
- Particles, $N = 8192$

SVGD parameters:

- Step size, $h = 0.05$
- Particles, $N = 8192$
- RBF kernel, $\sigma^2 = \mathrm{med}^2 / \log N$

Both algorithms use the same prior distribution and a batch size of 100.

Outline

Conclusion and future work

Conclusion

- Cubature allows one to simulate SDEs as a cloud of particles.
- When applied to the Langevin diffusion, cubature then gives a particle-based approach for approximate Bayesian inference.
- Cubature can outperform MCMC on a 2D Gaussian mixture and also performs well on a non-toy example, but is currently slow.
- Unlike SVGD, avoids $O(N^2)$ complexity in the number of particles.

Future Work

- Distribution compression (e.g. k-means clustering vs ball tree)
- Parallelism (e.g. JAX)
- Adaptive step sizes
- Different Markov processes
  (Hamiltonian dynamics, Riemannian Langevin dynamics, etc)

Thank you
for your attention!

Outline

References I

- G. A. Pavliotis. *Stochastic Processes and Applications*, Springer, New York, 2014.
- S. Vempala and A. Wibisono. *Rapid Convergence of the Unadjusted Langevin Algorithm: Isoperimetry Suffices*, NeurIPS 2019.
- N. Victoir. *Asymmetric Cubature Formulae with Few Points in High Dimension for Symmetric Measures*, SIAM Journal on Numerical Analysis, vol. 42, no. 1, pp. 209–227, 2005.
- T. Lyons and N. Victoir. *Cubature on Wiener space*, Proceedings of the Royal Society A, vol. 460, no. 2041, pp. 169–198, 2004.
- C. Litterer and T. Lyons. *High order recombination and an application to cubature on Wiener space*, Annals of Applied Probability, vol. 22, no. 4, pp. 1301–1327, 2012.
- M. Tchernychova. *Caratheodory cubature measures*, DPhil thesis, University of Oxford, 2016.

References II

- F. Cosentino, H. Oberhauser and A. Abate. *A randomized algorithm to reduce the support of discrete measures*, NeurIPS 2020.
- Y. Chen, M. Welling and A. Smola. *Super-Samples from Kernel Herding*, Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, 2010.
- R. Dwivedi and L. Mackey. *Kernel Thinning*, Proceedings of Thirty Fourth Conference on Learning Theory, 2021.
- M. Adachi, S. Hayakawa, M. Jørgensen, H. Oberhauser and M. Osborne. *Fast Bayesian Inference with Batch Bayesian Quadrature via Kernel Recombination*, NeurIPS 2022.
- M. Riabiz, W. Ye Chen, J. Cockayne, P. Swietach, S. Niederer, L. Mackey and C. Oates. *Optimal thinning of MCMC output*, Journal of Royal Statistical Society B (Statistical Methodology), 2022.

References III

- T. Eisner, B. Farkas, M. Haase and R. Nagel, *Operator Theoretic Aspects of Ergodic Theory*, Graduate Texts in Maths, Springer, 2015.
- D. Talay and L. Tubaro. *Expansion of the global error for numerical schemes solving stochastic differential equations*, Stochastic Analysis and Applications, vol. 8, no. 4, pp. 483–509, 1990.
- B. Leimkuhler, C. Matthews and M. Tretyakov. *On the long-time integration of stochastic gradient systems*, Proceedings of the Royal Society A, vol. 470, no. 2170, 2014.
- M. Lichman. *UCI machine learning repository*, https://archive.ics.uci.edu/ml, 2013. Binary covertype dataset: www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html
- D.-Y. Lim and S. Sabanis. *Polygonal Unadjusted Langevin Algorithms: Creating stable and efficient adaptive algorithms for neural networks*, arxiv:2105.13937, 2021.