Probabilistic Quadrature Rules: From Monte Carlo to Bayes

Shashank Sule

Amherst College

Decmeber 15, 2019

Quadrature

- How to compute

$$I = \int_{[0,1]} f(x)\mu(x)\,dx$$

  where μ satisfies the properties of a density function on $[0, 1]$?
- Example:

$$I = \int_{[0,1]} x^2\,dx$$

  $f(x) = x^2$ and $\mu(x) = \chi_{[0,1]}$
- Insight: Think *expected value*:

$$\mathbb{E}[f(X)] = \int_{[0,1]} f(x)\mu(x)\,dx \approx \langle f \rangle = \frac{1}{N}\sum_{i=1}^{N} f(X_i) \qquad (1)$$

- Thus, by computing the mean of N independently sampled values of $f(X_i)$ we can estimate the integral I. This is termed **Monte Carlo** integration.

Monte Carlo Strategies: Simple Monte Carlo

- Example: $f(x) = x^2$ and $\mu(x) = \chi_{[0,1]}$. Then $\frac{1}{N}\sum_{i=1}^{N} x_i^2 = \langle f \rangle$
- Generalization: By setting $\mu(x) = \frac{1}{\mathrm{Vol}(D)}\chi_D$ we can find $\int_D f(x)\,dx$ for any domain $D \subset \mathbb{R}^n$:

$$\int_D f(x)\,dx = \mathrm{Vol}(D)\int_D f(x)\frac{1}{\mathrm{Vol}(D)}\chi_D\,dx$$

- This is termed **Simple Monte Carlo**

Monte Carlo Strategies: Simple Monte Carlo

- Let's find the volume of the 3-dimensional L^1 (denoted $B_1^1(0)$) unit ball using SMC!

$$B_1^1(0) = \{x = (x_1, x_2, x_3) \in \mathbb{R}^3 \mid |x_1| + |x_2| + |x_3| \leq 1\}$$

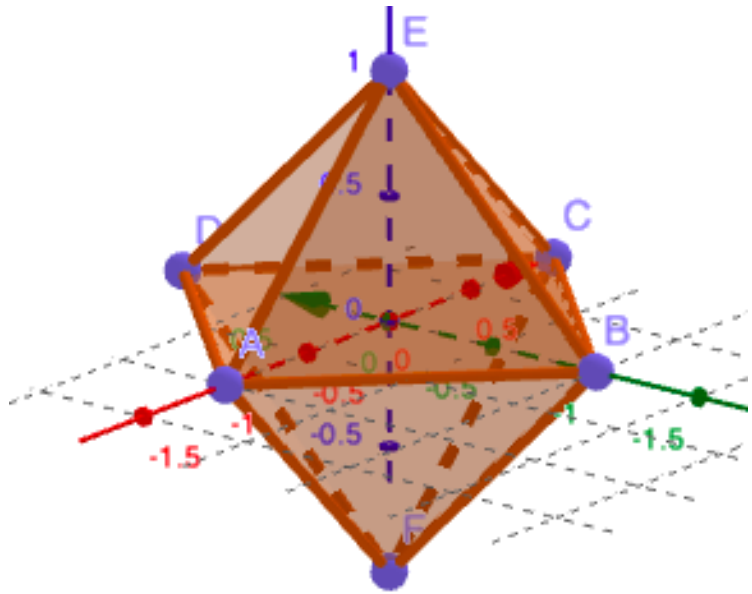


Figure: The L^1 unit ball in 3 dimensions

Monte Carlo Strategies: Simple Monte Carlo

- Note that

$$\mathrm{Vol}(B_1^1(0)) = \int_{B_1^1(0)} dx = 8 \int_{[-1,1]^3} \chi_{B_1^1(0)} \frac{1}{8} \chi_{[-1,1]}\, dx$$

- Sample N points in the unit cube in 3 dimensions and if the sampled point lies in the unit ball, record that as a 1, and if not record it as a 0. In the end, add up all the 1's and divide by N.

Monte Carlo Strategies: Simple Monte Carlo

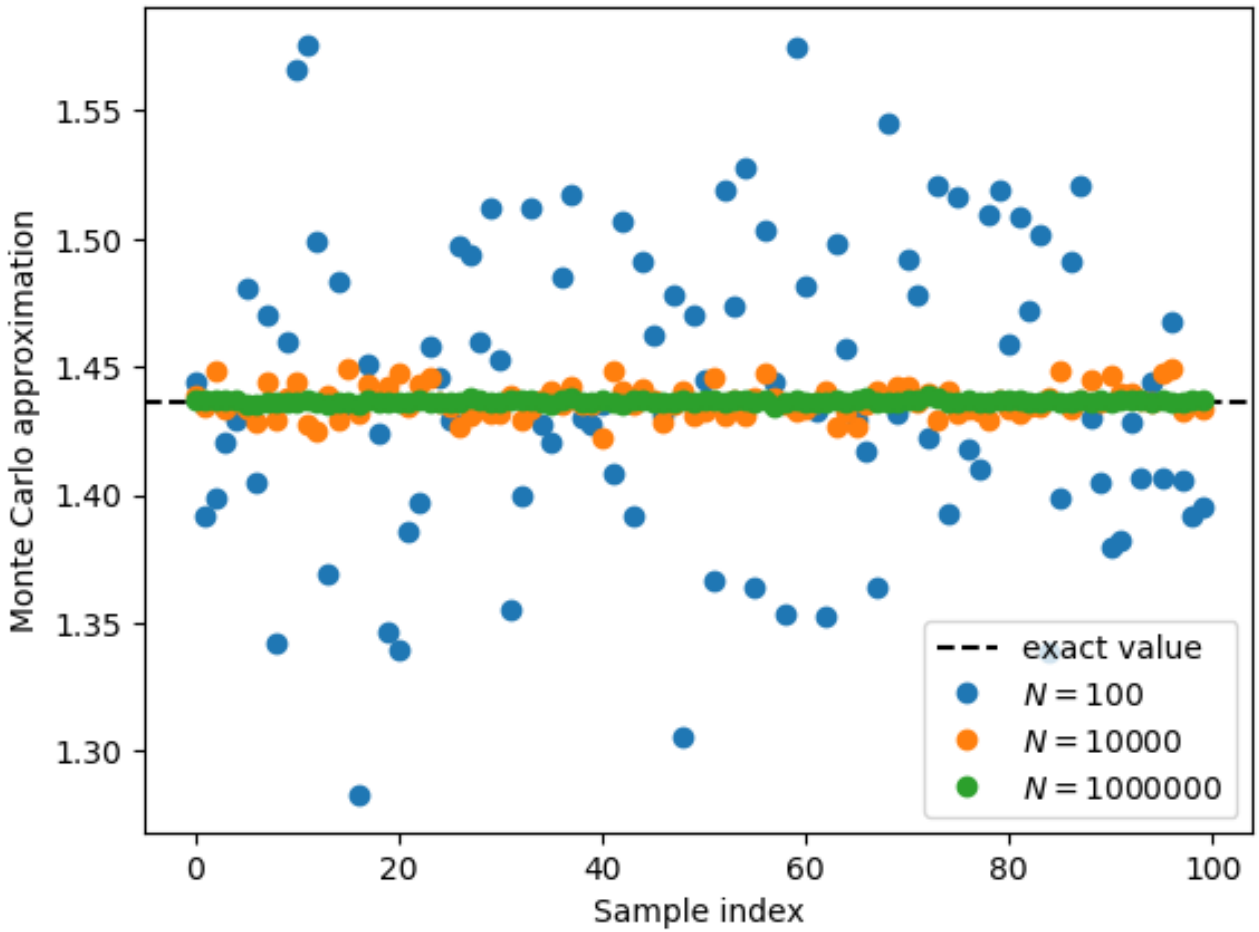


Figure: Computing $\mathrm{Vol}(B_1^1(0))$

Monte Carlo Strategies: Importance Sampling

- Note that the $N = 100$ case is particularly bad for SMC.
- This is because

$$Var(\langle f \rangle) = \frac{\text{Var}(f(X))}{N} \approx \mathcal{O}(N^{-1})$$

  the variance of the estimator depends on the variance of the underlying distribution X, which is uniform in SMC.
- Tweaking the distribution of X might improve the variance of the estimator and hence the accuracy of our results.
- This is termed **Importance Sampling**

Monte Carlo Strategies: Importance Sampling

- Let's compute
$$I = \int_{[0,1]} 2(1-x)e^x \, dx$$
- SMC: $f(x) = 2(1-x)e^x$ and $\mu(x) = \chi_{[0,1]}$.
- Importance Sampling: $f(x) = e^x$ and $\mu(x) = 2(1-x)$
- In the importance sampling case, the points will be sampled with a density of $2(1-x)$.

Monte Carlo Strategies: Importance Sampling

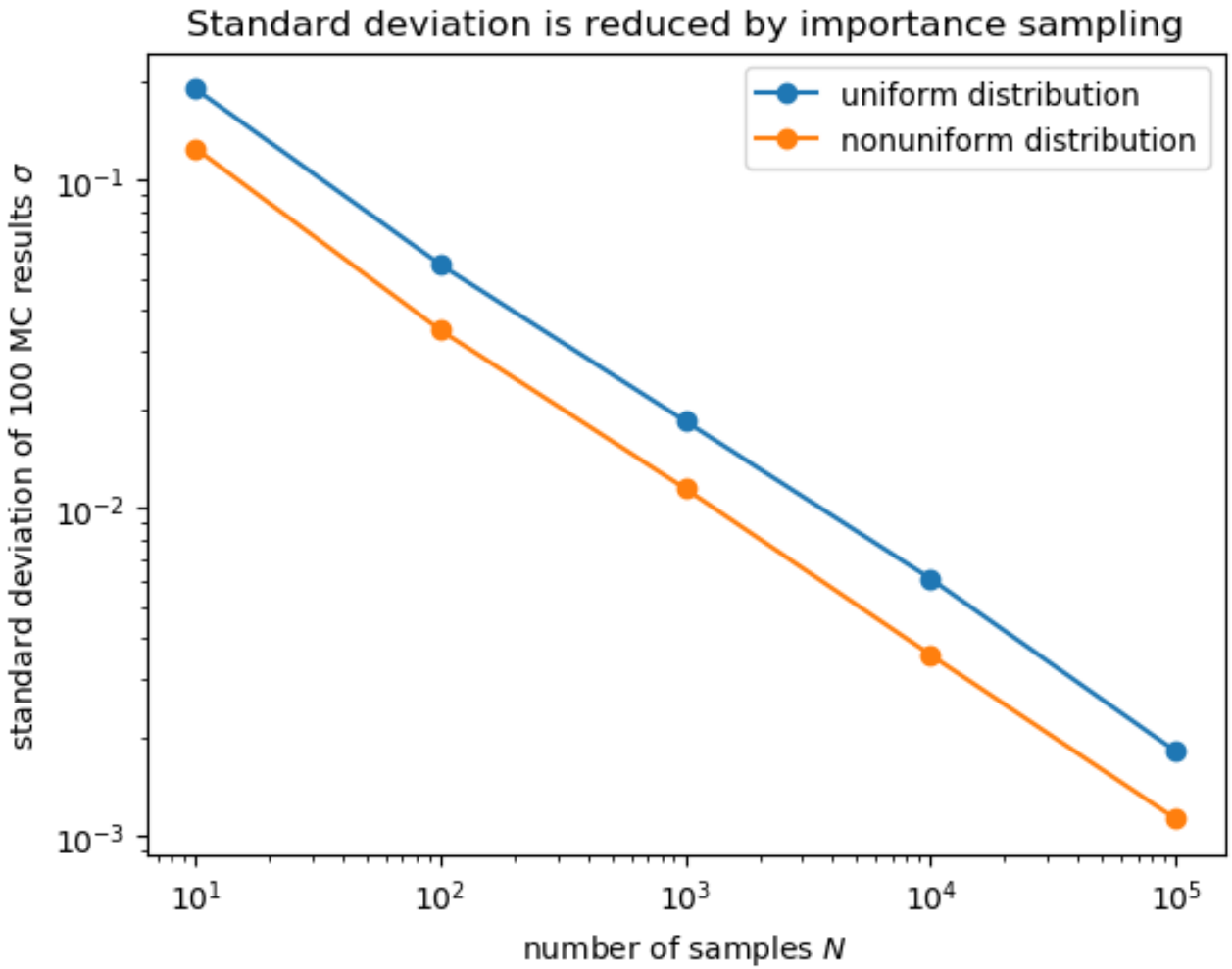


Figure: Importance Sampling improves accuracy!

Monte Carlo Strategies: Recursive Stratified Sampling

- Although IS improved the variance, the rate of convergence remained $1/N$.
- **Recursive Stratified Sampling** helps this problem.
- We divide the domain of integration, D into subdomains of equal size, termed A and B and compute I in the following way:

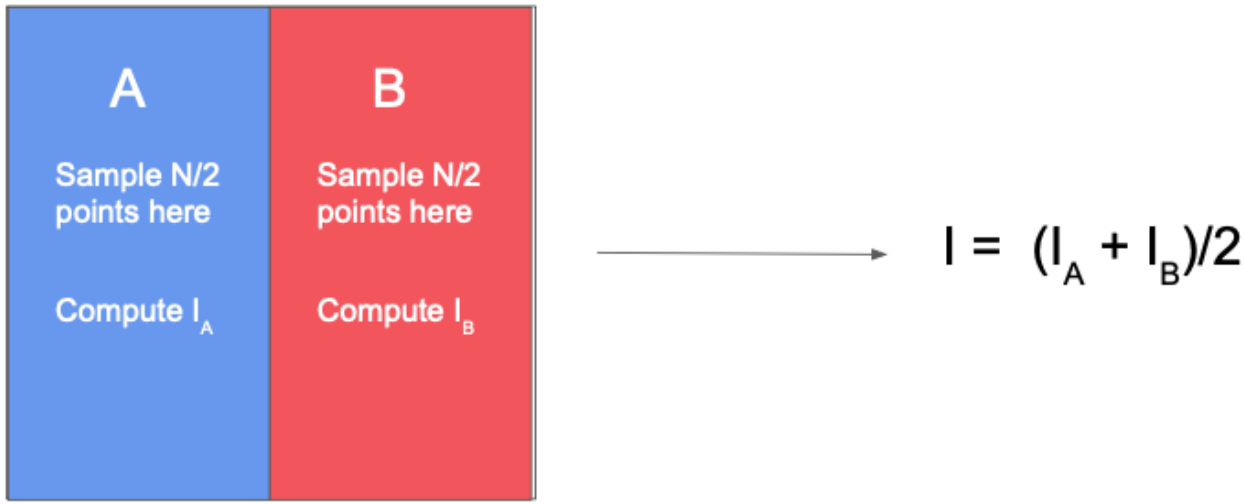


Figure: How RSS works

Recursive Stratified Sampling

- That the estimator $\langle f \rangle'$ can be viewed as a restricted case of the standard Monte Carlo estimator $\langle f \rangle$ where half the points are sampled independently from A and the other half is sampled independently from B:

$$I \approx \frac{1}{2}\left(\frac{1}{N/2}\sum_{i=1}^{N/2} f_A(X_i) + \frac{1}{N/2}\sum_{i=1}^{N/2} f_B(X_i)\right) = \langle f \rangle'$$

- **Parallel Axis theorem**: $\sigma^2(\langle f \rangle) \geq \sigma^2(\langle f \rangle')$

Monte Carlo Strategies: Recursive Stratified Sampling

Compute I by repeatedly subdividing D and computing $\langle f \rangle'$ at each step.
This is the **Recursive Stratified Sampling** algorithm:

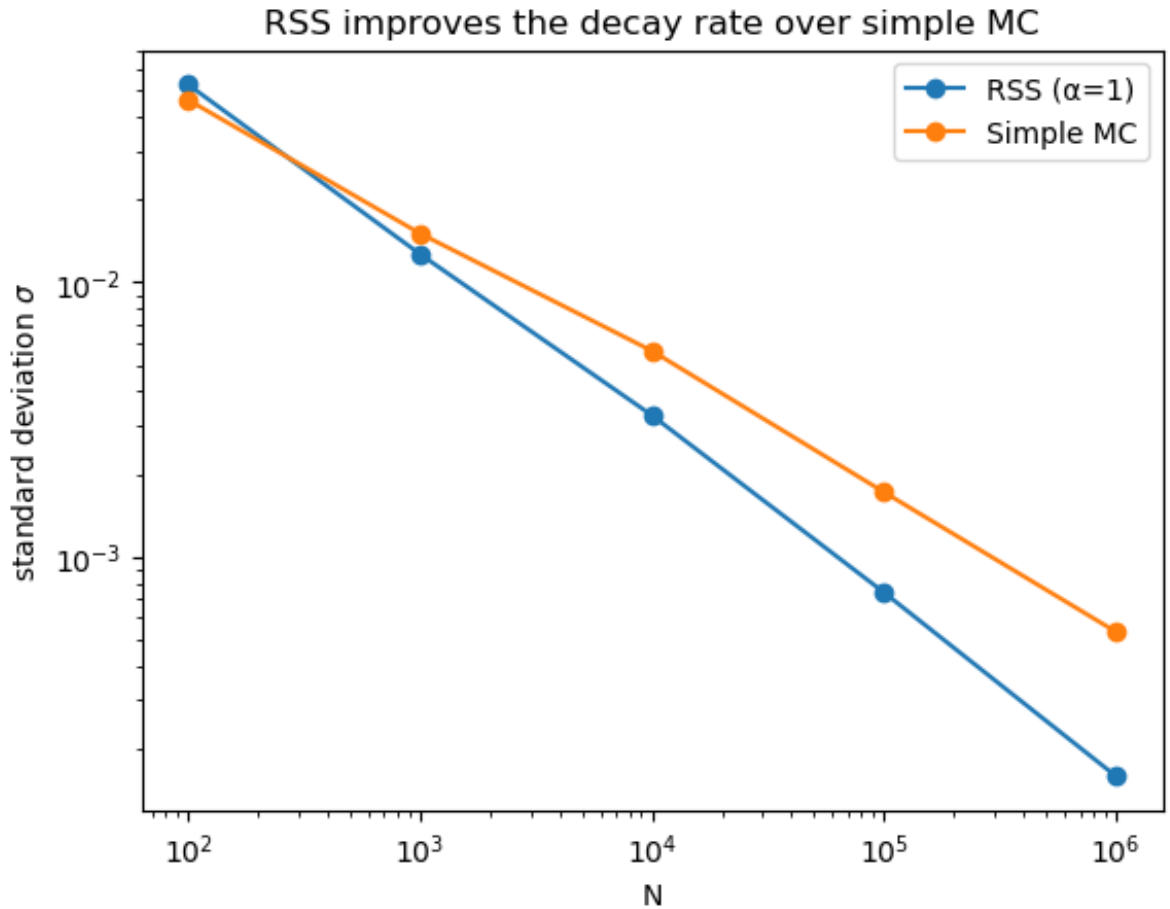


Figure: RSS improves the rate of convergence!

Monte Carlo algorithms: Summary

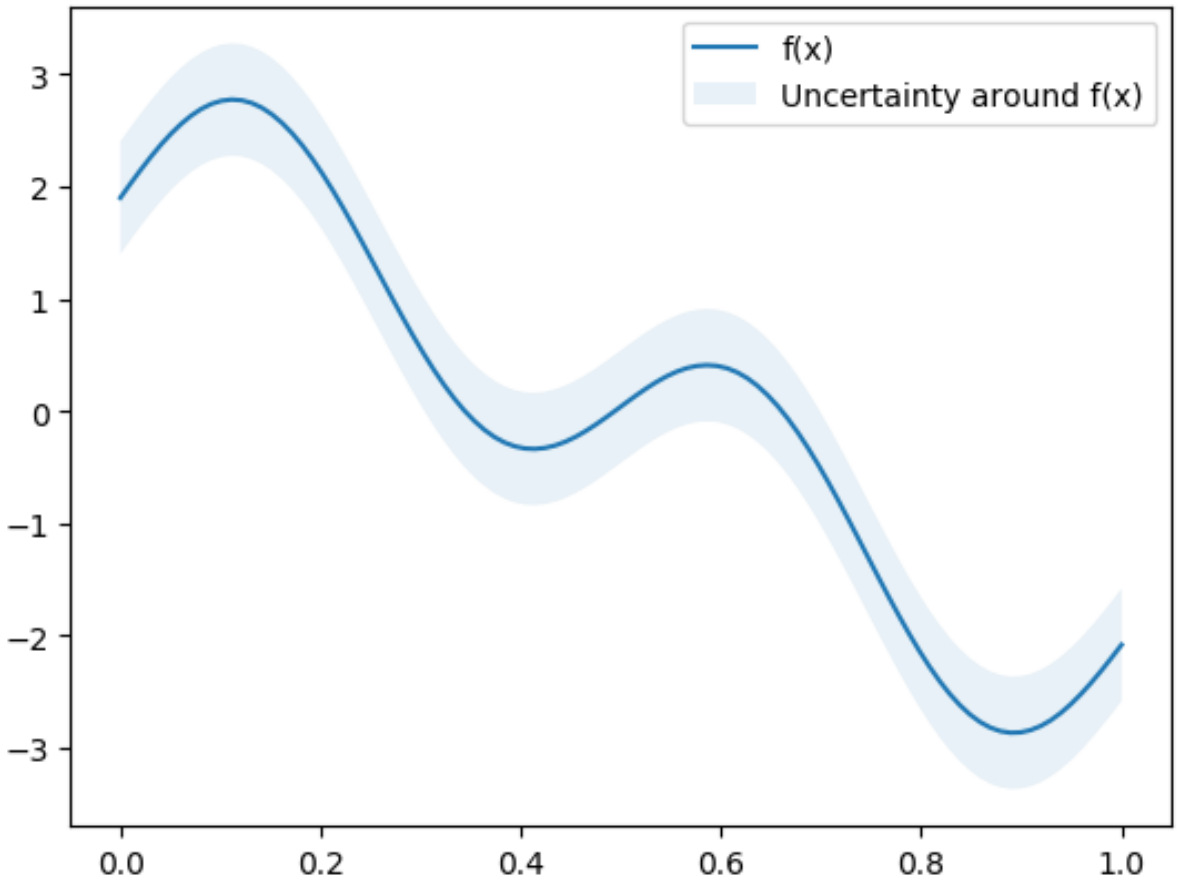
- The SMC, IS, and RSS algorithms are all modifications of the Monte Carlo estimator.
- Importance sampling maintains the $1/2$ rate of convergence, but improves the variance by a constant.
- RSS improves the rate of convergence: it is more desirable method when the function is cheap to sample.
- Despite these improvements, we still need about 1 million function samples to get to within 4 digits of the true answer
- The Bayesian estimator solves this accuracy problem dramatically.

Bayesian Quadrature

- The Bayesian strategy: I is itself random and depends on the observed values of f.
- There is inherent uncertainty in the value of $f(x)$. We need a prior on f to our beliefs on the value of $f(x)$.
- Let x_i be sampled independently from X and let $\mathbf{f} = [f(x_1) \dots f(x_n)]^\top$ be the vector of observed values of f. Then obtain a posterior on f, (termed $\hat{f}$) by conditioning on $\mathbf{f}$.
- Next, compute $\hat{I} \mid \mathbf{f}, \hat{f}$ where $\hat{I} = \int_{[0,1]} f(x)\mu(x)\,dx$. Note that I is a function of the posterior distribution!
- Provided a convenient prior on f we can use the expectation of $\hat{I}$ viewed as a distribution to estimate I. This is **Bayesian Quadrature**.

Gaussian Processes: Motivation

- A natural choice: $f(x) \sim N(m(x), \sigma(x))$



Gaussian Processes: Definition

- Let GP be a distribution on functions $f : [0, 1] \mapsto \mathbb{R}$. GP is termed a **Gaussian Process** when for any finite set of points $D = [x_1, \ldots x_n]$, the finite set of values of f, i.e $f(D) = [f(x_1) \ldots f(x_n)]^\top$ follows a joint normal distribution with mean μ and covariance matrix $k(D, D)$.
- Furthermore, μ and $k(D, D)$ entries of $k(D, D)$, termed $k(x, x')$ are given by a function called a *covariance kernel* satisfying symmetry, positivity, and 1-definiteness, i.e $k(x, x') = k(x', x)$, $k(x, y) > 0$, and $k(x, x) = 1$.
- Note that if $D = \{x\}$ then $f \sim N(\mu(x), k(x, x))$

Gaussian Processes: Example

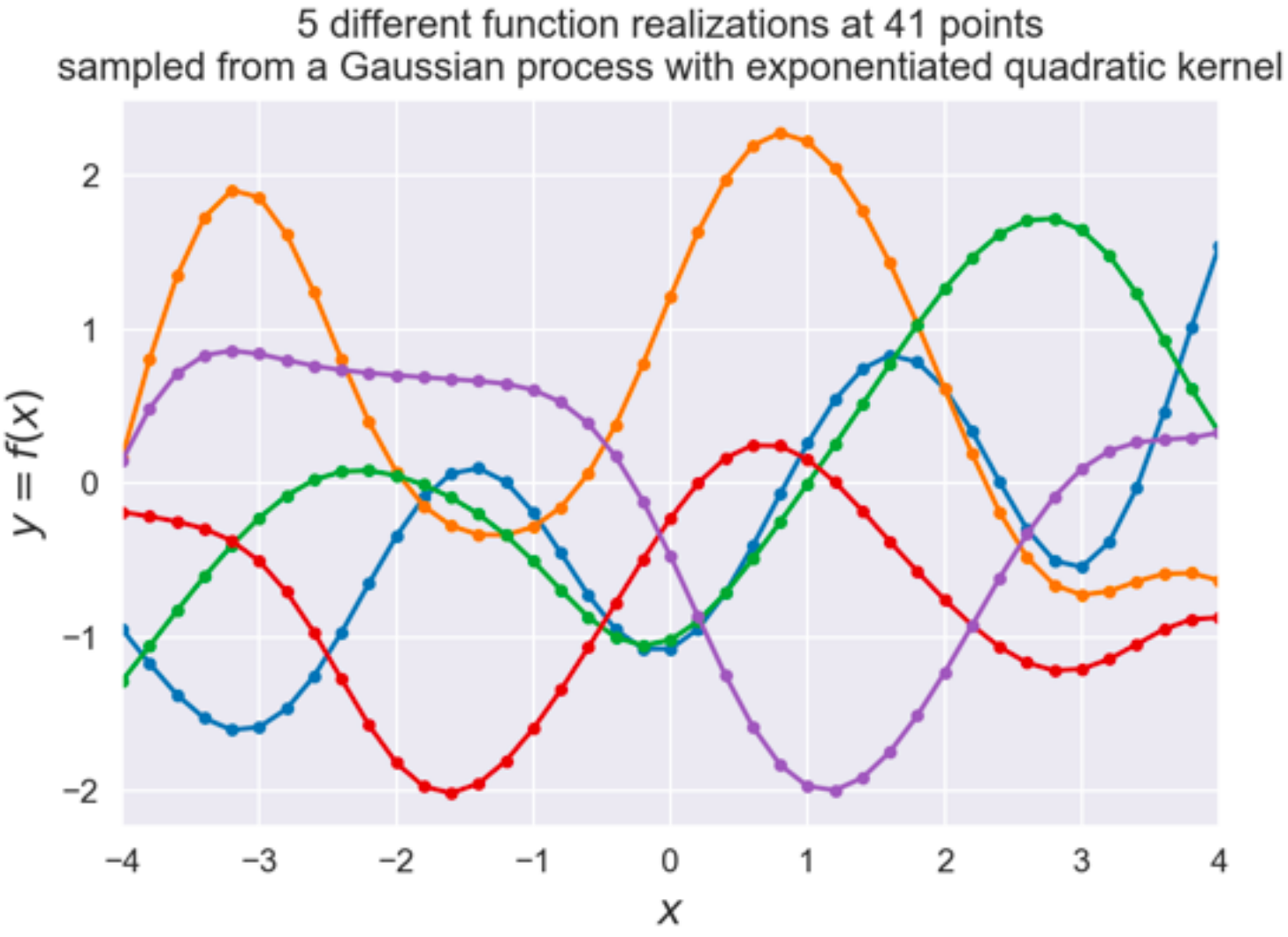


Figure: Illustrating a Gaussian process (Roelants n.d.)

Gaussian Processes: Theoretical results

Theorem (Roelants n.d.)

Let $f \sim GP$ and $f(D)$ be a set of finite evaluations of f on the set D. Then $f \mid \mathbf{f} \sim GP$

This result follows from the fact that the conditional of a joint normal conditioned on any subset of the variables is also a joint normal.

Corollary (Minka 2000)

Fix $x \in [0,1]$. Then

$$f(x) \mid \mathbf{f} \sim N(k(x,D)k(D,D)^{-1}f(D), k(x,x) - k(x,D)k(D,D)^{-1}k(D,x))$$

The above corollary gives an explicit formula for $f(x)$ as a posterior distribution.

Gaussian Processes: Theoretical Results (contd.)

Theorem (Commutativity of Expectations (Ghahramani and Rasmussen 2003))

Let $\bar{f}$ be the posterior mean of $f \mid f(D)$. Then

$$\mathbb{E}_{f|f(D)}[\mathbb{E}_X[f \mid f(D)]] = \mathbb{E}_X[\bar{f}]$$

Gaussian Processes: The posterior process

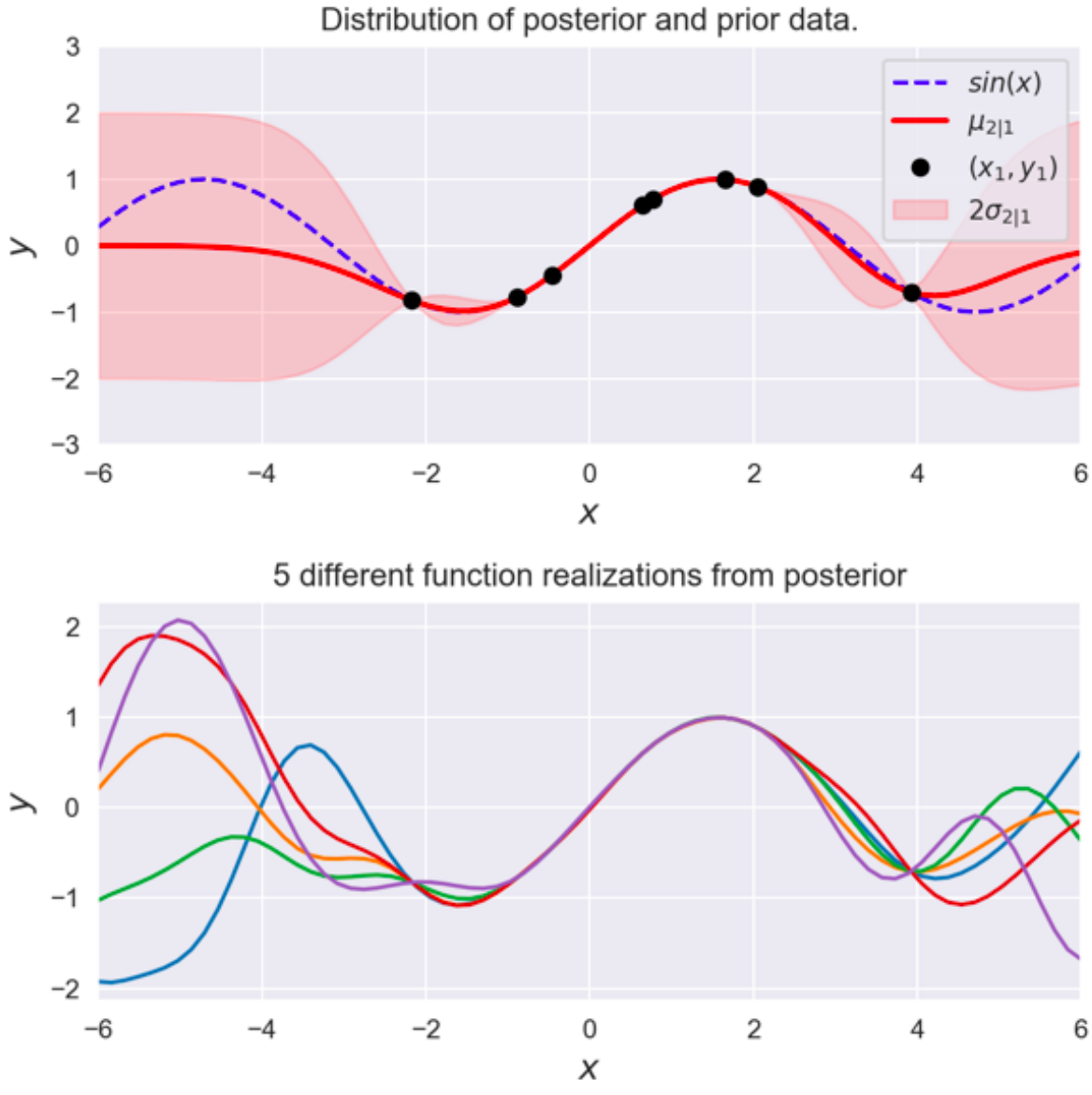


Figure: Sharpening beliefs from sampling (Roelants n.d.)

A simple Bayesian quadrature algorithm

- $I \approx \int_{[0,1]} m(x)\mu(x)\,dx$ where

$$m(x) = k(x, D)k(D, D)^{-1}f(D)$$

- Plugging in for m we have that

$$\hat{I} = (u(D))^{\top} k(D, D)^{-1}f(D)$$

  where $(u(D))^{\top} = \int_{[0,1]} k(x, D)\mu(x)\,dx$
- The variance of the estimator doesn't even depend on function values!
- However, to keep the algorithm probabilistic, we need not worry about variance minimzation.

A Simple Bayesian quadrature algorithm

- Let's compute $I = \int_{[0,1]} x^2 \, dx$ using Bayesian Quadrature!
- Pick k to be the **Lorentzian Kernel**, where

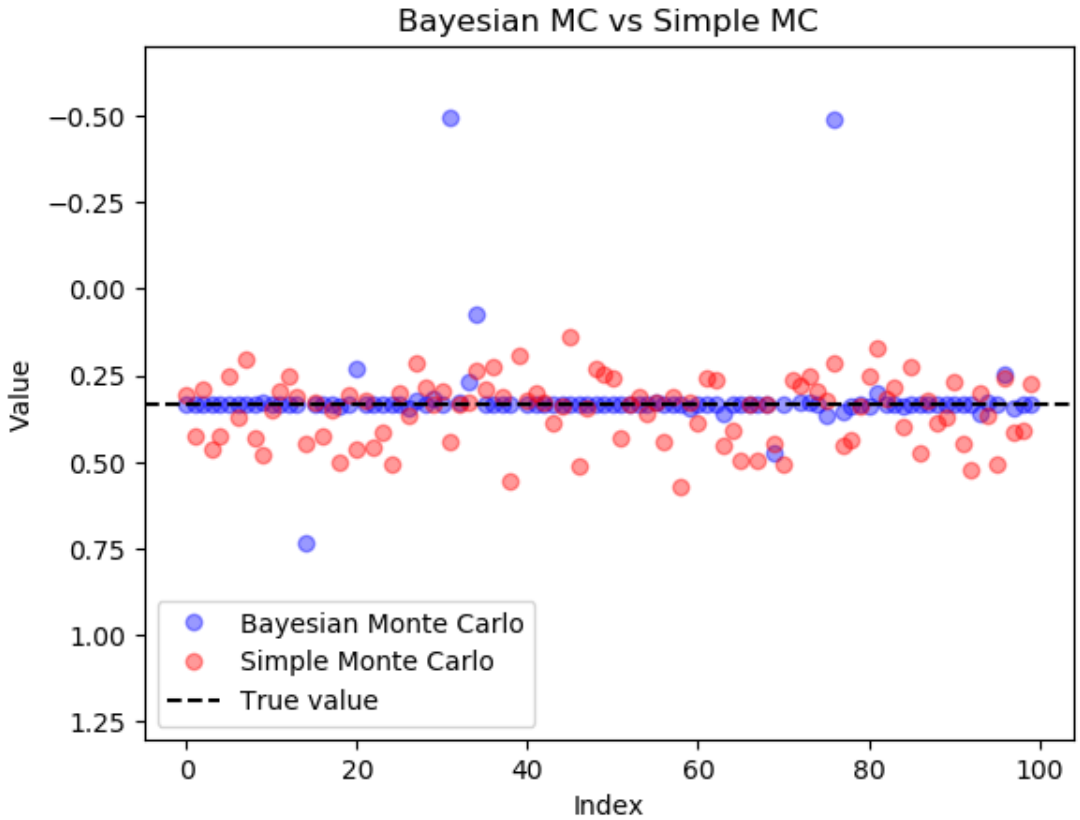
$$k(x, y) = \frac{1}{1 + |x - y|^2}$$

- $(u(D))^\top$ can be computed analytically when $\mu(x) \equiv 1$:

$$\begin{aligned}\int_{[0,1]} k(x, x_i)\mu(x) \, dx &= \int_{[0,1]} \frac{1}{1 + |x - x_i|^2} \, dx \\ &= \arctan(1 - x_i) - \arctan(-x_i) \\ &= \arctan(1 - x_i) + \arctan(x_i)\end{aligned}$$

A Simple Bayesian Quadrature algorithm

It takes only 10 samples for Bayesian Quadrature to get within 4 digits of the true answer while Monte Carlo will likely take more than 1000 samples. But BQ is also numerically unstable because it depends on inverting a matrix!



Conclusion

- Monte Carlo: numerically efficient and dimension-independent.
- The importance sampling and RSS algorithms are more accuracy-efficient implementations of the Monte Carlo estimator, yet not as accurate as Bayesian Quadrature.
- But BQ suffers from the curse of dimensionality and poor conditioning.
- Conclusion: There is no one algorithm that is best for all integrals!
- Acknowledgements: The Monte Carlo algorithms (Implemented in C) can be found in (Press et al. 1992).

References


Zoubin Ghahramani and Carl E. Rasmussen. "Bayesian Monte Carlo". In: (2003). Ed. by S. Becker, S. Thrun, and K. Obermayer, pp. 505–512. URL: http://papers.nips.cc/paper/2150-bayesian-monte-carlo.pdf.


Thomas P Minka. *Deriving quadrature rules from Gaussian processes*. Tech. rep. 2000.


William H Press et al. "Numerical recipes in C++". In: *The art of scientific computing* 2 (1992), p. 1002.


Peter Roelants. *Understanding Gaussian Processes*. URL: https://peterroelants.github.io/posts/gaussian-process-tutorial/.