Quasi-Monte Carlo for Functions of Multi-Dimensional Integrals

Aleksei G. Sorokin[1], Jagadeeswaran Rathinavel[2]

[1]Illinois Institute of Technology (IIT), Department of Applied Mathematics
[2]Wi-Tronix LLC

Motivating Examples

Problem: Given prior $P(\theta)$ and likelihood $P(Y \mid \theta)$, find posterior mean

$$\mathbb{E}[\theta \mid Y] \overset{\text{Bayes'}}{=} \frac{\int \theta P(Y \mid \theta) P(\theta) \mathrm{d}\theta}{\int P(Y \mid \theta) P(\theta) \mathrm{d}\theta} = \frac{\mu_1}{\mu_2} = C(\mu_1, \mu_2)$$

When posterior mean is unknown, approximate ratio/function of integrals
Other Problems written as functions of multiple integrals

- Vectorized expectation $\mathbb{E}[\boldsymbol{Y}] = \left(\mathbb{E}[Y_1], \mathbb{E}[Y_2] \ldots, \mathbb{E}[Y_k]\right)^T$
- $\text{Cov}(U,V) = \mathbb{E}[UV] - \mathbb{E}[U]\mathbb{E}[V]$
- Sensitivity indices for global sensitivity analysis

Question: How to extend single integral approximation to function(s) of integral(s)?

Framework

Problem: Approximate $C(\mathbb{E}[\boldsymbol{f}(\boldsymbol{X})])$ within error tolerance ε from error metric map $h(\cdot,\varepsilon)$, where $\boldsymbol{X} \sim \mathcal{U}[0,1]^d$, $\boldsymbol{f}$ a vector function, C a scalar function [1]

- Transformations can take a variety of integrals into this form [2]
- Example metrics maps
  - $h(s,\varepsilon) = \varepsilon_{\text{abs}}$, absolute error
  - $h(s,\varepsilon) = \max(\varepsilon_{\text{abs}}, |s|\varepsilon_{\text{rel}})$, absolute or relative
  - $h(s,\varepsilon) = \min(\varepsilon_{\text{abs}}, |s|\varepsilon_{\text{rel}})$, absolute and relative

Individual solutions

$$\boldsymbol{\mu} = (\mu_1,\dots,\mu_\rho)^T = (\mathbb{E}[f_1(\boldsymbol{X})],\dots\mathbb{E}[f_\rho(\boldsymbol{X})])^T$$

Combined solution

$$s = C(\boldsymbol{\mu}) = C(\mu_1,\dots,\mu_\rho)$$

Proposed Method

Ideas

- Quasi-Monte Carlo (QMC) methods can efficiently compute guaranteed bounds on $\boldsymbol{\mu}$ s.t. $\boldsymbol{\mu} \in [\boldsymbol{\mu}^-, \boldsymbol{\mu}^+]$ under assumptions on $\boldsymbol{f}$ [3, 4]
- Often straightforward to find bound propogation functions C^-, C^+ s.t.

$$s \in [s^-, s^+] = [C^-(\boldsymbol{\mu}^-, \boldsymbol{\mu}^+), C^+(\boldsymbol{\mu}^-, \boldsymbol{\mu}^+)]$$

- QMC methods iteratively double sample size

Method

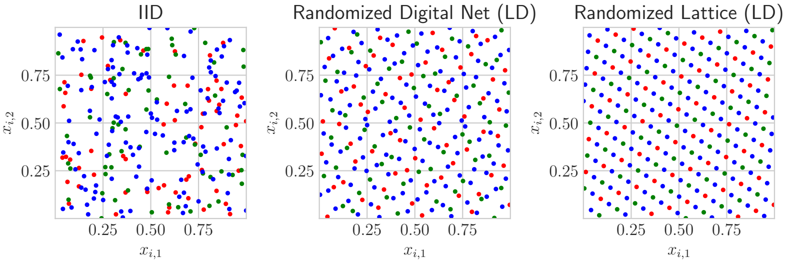
1. Sample $\boldsymbol{f}$ at n QMC samples $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n \in [0,1]^d$
2. Compute individual bounds $\boldsymbol{\mu}^-, \boldsymbol{\mu}^+$
3. Compute combined bounds s^-, s^+
4. If $s^+ - s^- \geq h(s^-, \varepsilon) + h(s^+, \varepsilon)$, set $n \leftarrow 2n$ and go to step 1
5. Compute optimal approximation $\hat{s} = \frac{1}{2}\left[s^- + s^+ + h(s^-, \varepsilon) - h(s^+, \varepsilon)\right]$

Quasi-Monte Carlo (QMC) Methods

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{f}(\boldsymbol{x}_i) \approx \int_{[0,1]^d} f(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \boldsymbol{\mu}$$

$(\boldsymbol{x}_i)_{i\geq 1} \subseteq [0,1]^d$ sampling nodes chosen to be

- IID $\rightarrow$ Crude Monte Carlo $\rightarrow$ $\mathcal{O}(n^{-1/2})$ convergence of $\hat{\boldsymbol{\mu}}$ to $\boldsymbol{\mu}$
- Low Discrepancy (LD) $\rightarrow$ QMC $\rightarrow$ $\mathcal{O}(n^{-1+\delta})$ convergence, $\forall \delta > 0$
  - Prefer extensible, randomized LD sequences with $n = 2^m$



Covariance Example

Combined Solution: $s = \text{Cov}(U,V) = \mathbb{E}[UV] - \mathbb{E}[U]\mathbb{E}[V] \in \mathbb{R}$

Individual Solutions: $\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} = \begin{pmatrix} \mathbb{E}[U] \\ \mathbb{E}[V] \\ \mathbb{E}[UV] \end{pmatrix} \in \mathbb{R}^3$

$$\begin{aligned} C(\boldsymbol{\mu}) &= \mu_3 - \mu_1\mu_2 \\ C^-(\boldsymbol{\mu}^-,\boldsymbol{\mu}^+) &= \mu_3^- - \max\left(\mu_1^+\mu_2^+,\, \mu_1^+\mu_2^-,\, \mu_1^-\mu_2^+,\, \mu_1^-\mu_2^-\right) \\ C^+(\boldsymbol{\mu}^-,\boldsymbol{\mu}^+) &= \mu_3^+ - \min\left(\mu_1^+\mu_2^+,\, \mu_1^+\mu_2^-,\, \mu_1^-\mu_2^+,\, \mu_1^-\mu_2^-\right) \end{aligned}$$

QMCPy [5] implementation requires specifying C^-, C^+

Vectorized Functions and Dependency

Up to now: $\boldsymbol{f} : [0,1]^d \to \mathbb{R}^\rho$; $\boldsymbol{\mu} \in \mathbb{R}^\rho$; $C^-, C^+ : \mathbb{R}^\rho \times \mathbb{R}^\rho \to \mathbb{R}$; $s \in \mathbb{R}$

Generalized: $\boldsymbol{f} : [0,1]^d \to \mathbb{R}^{\boldsymbol{\rho}}$; $\boldsymbol{\mu} \in \mathbb{R}^{\boldsymbol{\rho}}$; $\boldsymbol{C}^-, \boldsymbol{C}^+ : \mathbb{R}^{\boldsymbol{\rho}} \times \mathbb{R}^{\boldsymbol{\rho}} \to \mathbb{R}^{\boldsymbol{\eta}}$; $\boldsymbol{s} \in \mathbb{R}^{\boldsymbol{\eta}}$

$\boldsymbol{\rho}, \boldsymbol{\eta}$ shape vectors, e.g. $\boldsymbol{s} \in \mathbb{R}^{2 \times 3 \times 4} \implies \boldsymbol{\eta} = (2,3,4)^T$

Dependency function $D : \{\text{True}, \text{False}\}^{\boldsymbol{\eta}} \to \{\text{True}, \text{False}\}^{\boldsymbol{\rho}}$ answers question:

If $s_{\boldsymbol{i}}$ insufficiently approximated, which $\mu_{\boldsymbol{j}}$ require further sampling? $\equiv$
If $s_{\boldsymbol{i}}$ sufficiently approximated, which $\mu_{\boldsymbol{j}}$ can we ignore computing f for?

Multi-indices $\quad \boldsymbol{0} \le \boldsymbol{i} \le \boldsymbol{\eta}, \quad \boldsymbol{0} \le \boldsymbol{j} \le \boldsymbol{\rho}$

D enables economical function evaluation

Vectorized Acquisition Functions for Bayesian Optimization (BO)

BO sequentially optimizes a black-box function via surrogate Gaussian process g [6]

Batch sequential optimization: choose next q points to query the function s.t.

$$\boldsymbol{z}_{n+1},\dots,\boldsymbol{z}_{n+q} = \underset{\mathsf{Z}\in[0,1]^{q\times d}}{\operatorname{argmax}}\ \alpha(\mathsf{Z})$$

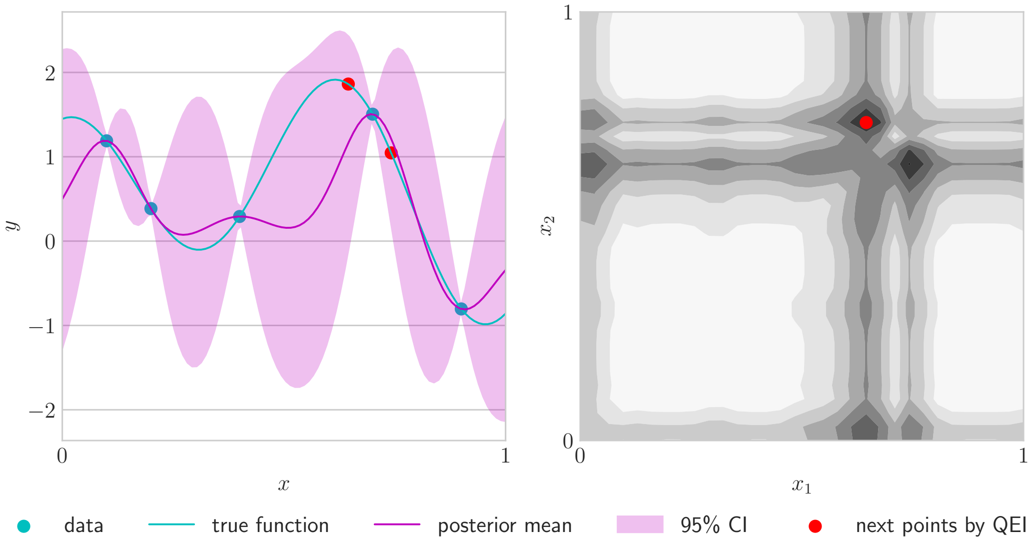
q-EI acquisition function with Gaussian posterior $\boldsymbol{Y} \sim P(g(\mathsf{Z}) \mid D)$ and current best y^*

$$\alpha(\mathsf{Z}) = \mathbb{E}[a(\boldsymbol{Y})|\boldsymbol{Y} \sim P(g(\mathsf{Z})|D)], \qquad a(\boldsymbol{y}) = \max_{1\le i\le q}[\max(y_i - y^*, 0)]$$

Vectorize computation at candidate batches $\mathsf{Z}_1,\dots,\mathsf{Z}_k \in [0,1]^{q\times d}$

$$\begin{pmatrix}\alpha(\mathsf{Z}_1)\\ \vdots \\ \alpha(\mathsf{Z}_k)\end{pmatrix} = \mathbb{E}\begin{pmatrix}a\left(\Phi_1^{-1}(\boldsymbol{X})\right)\\ \vdots \\ a\left(\Phi_k^{-1}(\boldsymbol{X})\right)\end{pmatrix}, \qquad \boldsymbol{X} \sim \mathcal{U}[0,1]^q$$

BO via q-EI, $q = 2$


2
1
0
−1
−2
y
0
1
x
1
0
x2
0
1
x1
data
true function
posterior mean
95% CI
next points by QEI

Sensitivity Indices [7, 8]: quantify variance attributable to $u \subseteq \{1,\dots,d\}$

Funcational ANOVA decomposes $f \in L^2(0,1)^d$ into orthogonal $\{f_u\}_{u \subseteq \{1,\dots,d\}}$ s.t.

$$f(\boldsymbol{x}) = \sum_{u \subseteq \{1,\dots,d\}} f_u(\boldsymbol{x}_u) \qquad \text{and} \qquad \sigma^2 = \sum_{u \subseteq \{1,\dots,d\}} \sigma_u^2$$

Closed and total Sobol' indices

$$\underline{\tau}_u^2 = \sum_{v \subset u} \sigma_v^2 \qquad \text{and} \qquad \overline{\tau}_u^2 = \sum_{v \cap u \neq \emptyset} \sigma_v^2$$

Closed and total sensitivity indices (normalized Sobol' indices)

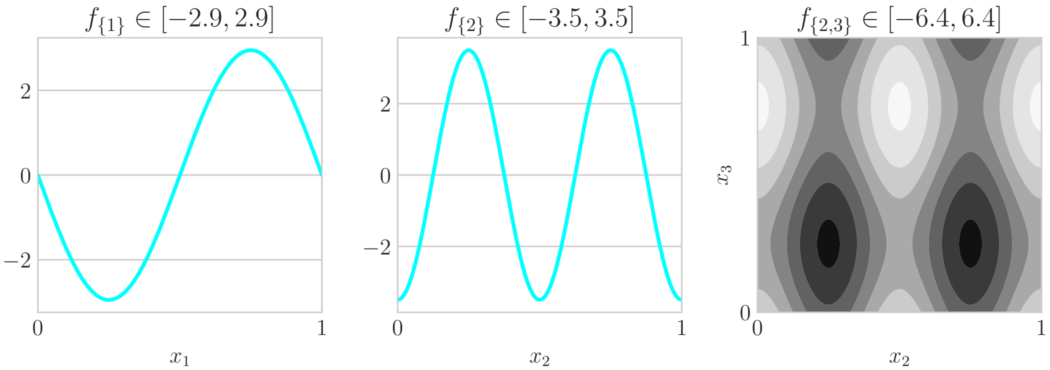
$$\underline{s}_u = \underline{\tau}_u^2 / \sigma^2 \qquad \text{and} \qquad \overline{s}_u = \overline{\tau}_u^2 / \sigma^2$$

$\underline{s}_u$ and $\overline{s}_u$ can be written in terms of 6 total expectations

Sensitivity Indices: Ishigami Function [9]

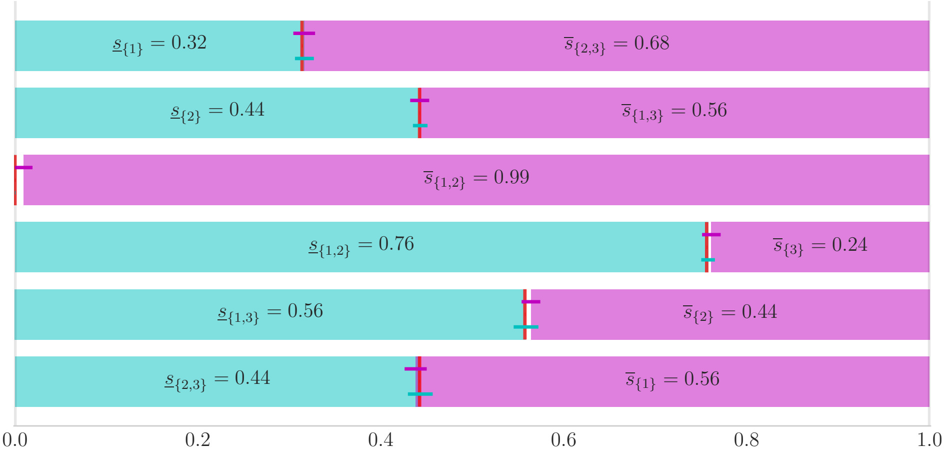
$$g(\boldsymbol{T}) = \left(1 + bT_3^4\right)\sin(T_1) + a\sin^2(T_2), \qquad \boldsymbol{T} \sim \mathcal{U}(-\pi,\pi)^3$$

$$f(\boldsymbol{X}) = g\left(\pi\left(2\boldsymbol{X}-1\right)\right), \qquad \boldsymbol{X} \sim \mathcal{U}(0,1)^3$$



Sensitivity Indices: Ishigami Function

$$\underline{s}_u + \overline{s}_{u^c} = 1, \qquad u \in \{1, \dots, d\}$$



Sensitivity Indices: Neural Network Iris Classifier [10]

Trained network achieves 98% accuracy on validation set

Singleton Closed Indices

	sepal length	sepal width	petal length	petal width	sum
setosa	0.2%	5.9%	71.4%	4.6%	82.0%
versicolor	7.1%	2.2%	32.8%	2.1%	44.3%
virginica	8.2%	1.0%	50.0%	12.0%	71.2%

Petal length accounts for most variation among singletons

Non-singleton interactions, $|u| > 1$, are most important for differentiating versicolor

individual solutions: $\boldsymbol{\mu} \in \mathbb{R}^{6\times 4\times 3}$ combined solutions: $\boldsymbol{s} \in \mathbb{R}^{2\times 4\times 3}$

Neural Network Sensitivity Indices Code

```
data = load_iris()
target_names = data["target_names"]
xt,xv,yt,yv = train_test_split(data["data"],data["target"])
mplc = MLPClassifier(max_iter=1024).fit(xt,yt)
yhat = mplc.predict(xv)
print("accuracy: %.1f%%"%(100*(yv==yhat).mean()))

sampler = DigitalNetB2(dimension=4)
true_measure =  Uniform(sampler,xt.min(0),xt.max(0))
fun = CustomFun(true_measure,
    g = lambda x,compute_flags: mplc.predict_proba(x),
    dprime = (3,))
si_fun = SensitivityIndices(fun,indices="all")
qmc_algo = CubQMCNetG(si_fun,abs_tol=.005)
nn_sis,nn_sis_data = qmc_algo.integrate()
print(nn_sis_data)
```

Future Work

- Extend to QMC probabilistic error bounds on individual solutions [7, 11, 12]
- Support for adaptive IID Monte Carlo algorithms [13]
- Support for adaptive multilevel Monte Carlo algorithms [14, 15]
- Support combining individual bounds from QMC, IID Monte Carlo, and/or multilevel Monte Carlo stopping criterion
- Make multi-dimensional function and true measure construction more flexible in QMCPy

Thank you!

References I

1. Hickernell, F. J., Jiménez Rugama, L. A. & Li, D. in *Contemporary Computational Mathematics-A Celebration of the 80th Birthday of Ian Sloan* 597–619 (Springer, 2018).
2. Choi, S.-C. T., Hickernell, F. J., Jagadeeswaran, R., McCourt, M. J. & Sorokin, A. G. *Quasi-Monte Carlo Software.* in *Monte Carlo and Quasi-Monte Carlo Methods* (ed Keller, A.) (Springer International Publishing, Cham, 2022), 23–47. ISBN: 978-3-030-98319-2.
3. Jiménez Rugama, L. A. & Hickernell, F. J. *Adaptive Multidimensional Integration Based on Rank-1 Lattices.* 2014. arXiv: 1411.1966 [math.NA].
4. Hickernell, F. J. & Jiménez Rugama, L. A. *Reliable Adaptive Cubature Using Digital Sequences.* 2014. arXiv: 1410.8615 [math.NA].

References II

5. Choi, S.-C. T., Hickernell, F. J., Jagadeeswaran, R., McCourt, M. J. & Sorokin, A. G. *QMCPy: A Quasi-Monte Carlo Python Library.* 2020+. https://github.com/QMCSoftware/QMCSoftware.

6. Frazier, P. I. *A Tutorial on Bayesian Optimization.* 2018. https://arxiv.org/abs/1807.02811.

7. Owen, A. B. *Monte Carlo theory, methods and examples.* (2018).

8. Jiménez Rugama, L. A. & Gilquin, L. Reliable error estimation for Sobol' indices. *Statistics and Computing* **28,** 725–738. https://hal.inria.fr/hal-01358067 (July 2018).

9. Ishigami, T. & Homma, T. *An importance quantification technique in uncertainty analysis for computer models.* in *[1990] Proceedings. First International Symposium on Uncertainty Modeling and Analysis* (1990), 398–403.

References III

10. Hoyt, C. & Owen, A. B. Efficient Estimation of the ANOVA Mean Dimension, with an Application to Neural Net Classification. *SIAM/ASA Journal on Uncertainty Quantification* **9,** 708–730. https://doi.org/10.1137/20M1350236 (2021).
11. Rathinavel, J. & Hickernell, F. J. Fast automatic Bayesian cubature using lattice sampling. *Statistics and Computing* **29,** 1215–1229. ISSN: 1573-1375. http://dx.doi.org/10.1007/s11222-019-09895-9 (Sept. 2019).
12. Rathinavel, J. *Fast automatic Bayesian cubature using matching kernels and designs.* PhD Thesis (Illinois Institute of Technology, Chicago, Dec 2019). www.math.iit.edu.
13. Hickernell, F. J., Jiang, L., Liu, Y. & Owen, A. *Guaranteed Conservative Fixed Width Confidence Intervals Via Monte Carlo Sampling.* 2012. arXiv: 1208.4318 [math.ST].

References IV

14. Giles, M. Multilevel Monte Carlo Path Simulation. *Operations Research* **56,** 607–617 (June 2008).

15. Giles, M. B. & Waterhouse, B. J. Multilevel quasi-Monte Carlo path simulation. *Advanced Financial Modelling, Radon Series on Computational and Applied Mathematics* **8,** 165–181 (2009).