Monte Carlo with QMCPy for Vector Functions of Integrals

Aleksei G. Sorokin

Illinois Institute of Technology (IIT), Department of Applied Mathematics

Monte Carlo Problem

$$\text{True Mean} = \mu = \mathbb{E}[g(T)] = \mathbb{E}[f(X)] = \int_{[0,1]^d} f(x)\mathrm{d}x$$

- T, original random variable e.g. $T \sim \mathcal{N}(0, I)$
- $g : \mathcal{T} \to \mathbb{R}$, original integrand
- $X \sim \mathcal{U}[0,1]^d$, transformed random variable, e.g. $T \sim \Phi^{-1}(X)$
- $f : [0,1]^d \to \mathbb{R}$, transformed integrand

(Quasi-)Monte Carlo Method

$$\text{Sample Mean} = \hat{\mu} = \frac{1}{n}\sum_{i=0}^{n-1} f(X_i)$$

$$X_0, \ldots, X_{n-1} \sim \mathcal{U}[0,1]^d$$

	Crude Monte Carlo	**Quasi-Monte Carlo**
$X_0, \ldots, X_{n-1}$	independent identically distributed (IID)	low discrepancy (LD)
	gaps and clusters	even coverage
Rate of $\hat{\mu}$ to μ	$\mathcal{O}(n^{-1/2})$	$\mathcal{O}(n^{-1+\delta})$, any $\delta > 0$
Generally	slower	faster
but works for	more integrands	fewer integrands (provably)

QMCPy Software Components

Discrete Distribution generates sampling locations $X_0, X_1, \ldots$

True Measure defines T, facilitates automatic transform from g to f so

$$\mu = \mathbb{E}[g(T)] = \mathbb{E}[f(X)]$$

Integrand defines original g, stores transformed f

Stopping Criterion adaptively increases n until approximation

$$\hat{\mu} = \frac{1}{n} \sum_{i=0}^{n-1} f(X_i)$$

has error below user-defined tolerance ε

$$|\mu - \hat{\mu}| < \varepsilon$$

Vectorized Stopping Criteria

Quantity of interest s is now a function of multiple expectations / integrals

$$s = C(\mu_0, \mu_1, \dots)$$

Covariance of random variables Y and Z

$$\text{Cov}[Y, Z] = \mathbb{E}[YZ] - \mathbb{E}[Y]\mathbb{E}[Z] = \mu_0 - \mu_1\mu_2$$

Posterior Mean from prior $\varrho(\theta)$ and likelihood $\varrho(y|\theta)$

$$\mathbb{E}[\Theta|y] = \frac{\int \theta \varrho(y|\theta)\varrho(\theta)\mathrm{d}\theta}{\int \varrho(y|\theta)\varrho(\theta)\mathrm{d}\theta} = \frac{\mu_0}{\mu_1}$$

Sensitivity Indices for quantifying parameter importance

Enable adaptive approximation $\hat{s}$ of s where $|s - \hat{s}| < \varepsilon$

References

- Accompanying demo https://tinyurl.com/QMCPyPyDataChi2023
- QMCPy homepage https://qmcpy.org
- QMCPy article[1]
- Accessible introduction to Monte Carlo[2], discusses sensitivity indices in appendix

[1] Sou-Cheng T. Choi et al. "Quasi-Monte Carlo Software". In: *Monte Carlo and Quasi-Monte Carlo Methods*. Ed. by Alexander Keller. Cham: Springer International Publishing, 2022, pp. 23–47. ISBN: 978-3-030-98319-2.

[2] Art B. Owen. *Monte Carlo theory, methods and examples*. 2018.