Consistency of Markov chain quasi-Monte Carlo on continuous state spaces

Josef Dick
joint work with S. Chen and A. Owen

School of Mathematics and Statistics, UNSW, Sydney, Australia

The task is to approximate an integral

$$I_s(f) = \int_\Omega f(\boldsymbol{z})\pi(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}$$

for some integrand f, domain $\Omega \subseteq \mathbb{R}^s$, and density function π by some Monte-Carlo type quadrature rule

$$Q_{N,s}(f) = \frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{x}_n)$$

at some sample points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \Omega$.
Need to generate sample points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ with distribution π.

- Generate random numbers $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots \sim U(0,1)^d$.

- Generate random numbers $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots \sim U(0,1)^d$.
- Use transformation $\Psi : (0,1)^d \mapsto \Omega$ such that $\Psi(\boldsymbol{u}_i) \sim \pi$.

- Generate random numbers $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots \sim U(0,1)^d$.
- Use transformation $\Psi : (0,1)^d \mapsto \Omega$ such that $\Psi(\boldsymbol{u}_i) \sim \pi$.

Ψ may be obtained from

- Generate random numbers $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots \sim U(0,1)^d$.
- Use transformation $\Psi : (0,1)^d \mapsto \Omega$ such that $\Psi(\boldsymbol{u}_i) \sim \pi$.

Ψ may be obtained from

- Inversion of CDF

- Generate random numbers $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots \sim U(0,1)^d$.
- Use transformation $\Psi : (0,1)^d \mapsto \Omega$ such that $\Psi(\boldsymbol{u}_i) \sim \pi$.

Ψ may be obtained from

- Inversion of CDF
- Metropolis-Hastings Algorithm

Metropolis-Hastings Algorithm

For $\boldsymbol{x} \in \Omega$ let $\psi_{\boldsymbol{x}} : [0,1]^{d-1}$ be a generator for the transition kernel $P(\boldsymbol{x} \to \boldsymbol{y})$ with conditional density $p(\cdot \mid \boldsymbol{x})$. The Metropolis-Hastings sampler has

$$\phi(\boldsymbol{x}, \boldsymbol{u}) = \begin{cases} \boldsymbol{y}(\boldsymbol{x}, \boldsymbol{u}), & u_d \le A(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{x}, & u_d > A(\boldsymbol{x}, \boldsymbol{u}) \end{cases}$$

where $\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{u}) = \psi_{\boldsymbol{x}}(\boldsymbol{u}_{1:(d-1)})$ and

$$A(\boldsymbol{x}, \boldsymbol{u}) = \min\left(1, \frac{\pi(\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{u}))\, p(\boldsymbol{x} \mid \boldsymbol{y}(\boldsymbol{x}, \boldsymbol{u}))}{\pi(\boldsymbol{x})\, p(\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{u}) \mid \boldsymbol{x})}\right).$$

- Generate random numbers $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots \sim U(0,1)^d$.
- Use transformation $\Psi : (0,1)^d \mapsto \Omega$ such that $\Psi(\boldsymbol{u}_i) \sim \pi$.

Ψ may be obtained from

- Inversion of CDF
- Metropolis-Hastings Algorithm
- Gibbs sampler

Gibbs sampler

To construct the systematic scan Gibbs sampler let $\psi_{j,\boldsymbol{x}_{-j}}(\boldsymbol{u}_j)$ be a k_j–dimensional generator of the full conditional distribution of $x_j \in \mathbb{R}^{k_j}$ given x_ℓ for all $\ell \neq j$. This Gibbs sampler generates the new point using $\boldsymbol{u} \in [0,1]^d$ where $d = \sum_{j=1}^{s} k_j$. Write $\boldsymbol{u} = (\boldsymbol{u}_1, \dots, \boldsymbol{u}_s)$.
The systematic scan Gibbs sampler has

$$\phi(\boldsymbol{x}, \boldsymbol{u}) = (\phi_1(\boldsymbol{x}, \boldsymbol{u}), \phi_2(\boldsymbol{x}, \boldsymbol{u}), \dots, \phi_s(\boldsymbol{x}, \boldsymbol{u}))$$

where

$$\phi_j(\boldsymbol{x}, \boldsymbol{u}) = \psi_{j,\boldsymbol{x}_{[j]}}(\boldsymbol{u}_j)$$

and $\boldsymbol{x}_{[j]} = (\phi_1(\boldsymbol{x}, \boldsymbol{u}), \dots, \phi_{j-1}(\boldsymbol{x}, \boldsymbol{u}), x_{j+1}, \dots, x_d)$.

- Generate random numbers $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots \sim U(0,1)^d$.
- Use transformation $\Psi : (0,1)^d \mapsto \Omega$ such that $\Psi(\boldsymbol{u}_i) \sim \pi$.

Ψ may be obtained from

- Inversion of CDF
- Metropolis-Hastings Algorithm
- Gibbs sampler
- Inverse slice sampler

Inverse slice sampler

Let π be a probability density function on $\Omega \subset \mathbb{R}^s$. Let $\Omega' = \{(y, \boldsymbol{x}) \mid \boldsymbol{x} \in \Omega, 0 \le y \le \pi(\boldsymbol{x})\} \subset \mathbb{R}^{s+1}$ and let π' be the uniform distribution on Ω'. The inversive slice sampler is the systematic scan Gibbs sampler for π' with each $k_j = 1$ using inversion for every $\psi_{j,\boldsymbol{x}_{[j]}}$.

- Generate random numbers $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots \sim U(0,1)^d$.
- Use transformation $\Psi : (0,1)^d \mapsto \Omega$ such that $\Psi(\boldsymbol{u}_i) \sim \pi$.

Ψ may be obtained from

- Inversion of CDF
- Metropolis-Hastings Algorithm
- Gibbs sampler
- Inverse slice sampler
- ...

- Theory assumes that driver sequence $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ is random.

- Theory assumes that driver sequence $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ is random.
- But usually pseudo-random numbers are used.

- Theory assumes that driver sequence $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ is random.
- But usually pseudo-random numbers are used.
- Pseudo-random numbers are deterministically generated to **appear** random.

- Theory assumes that driver sequence $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ is random.
- But usually pseudo-random numbers are used.
- Pseudo-random numbers are deterministically generated to **appear** random.
- *Pseudo-random numbers cannot pass every conceivable statistical test. (Niederreiter)*

- Theory assumes that driver sequence $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ is random.
- But usually pseudo-random numbers are used.
- Pseudo-random numbers are deterministically generated to **appear** random.
- *Pseudo-random numbers cannot pass every conceivable statistical test. (Niederreiter)*
- Which properties should the pseudo-random numbers then satisfy if used in a MCMC algorithm in order to obtain the correct result? (Consistency)

- Theory assumes that driver sequence $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ is random.
- But usually pseudo-random numbers are used.
- Pseudo-random numbers are deterministically generated to **appear** random.
- *Pseudo-random numbers cannot pass every conceivable statistical test. (Niederreiter)*
- Which properties should the pseudo-random numbers then satisfy if used in a MCMC algorithm in order to obtain the correct result? (Consistency)
- If we know the property, can the pseudo-random numbers be improved? (Convergence)

Previous Work: Chentsov [1967], Owen and Tribble [2005], and Tribble and Owen [2008] showed consistency if Ω is finite set.

Numerical Results: Liao [1998], Chaudary [2004], Tribble [2007]

Monte Carlo vs. Quasi-Monte Carlo

Simple example:

Monte Carlo vs. Quasi-Monte Carlo

Simple example: $f : [0,1] \mapsto \mathbb{R}$. Estimate $\int_0^1 f(x)\,\mathrm{d}x$.

Monte Carlo vs. Quasi-Monte Carlo

Simple example: $f : [0,1] \mapsto \mathbb{R}$. Estimate $\int_0^1 f(x)\,\mathrm{d}x$.

Conclusion: Generate points uniformly.

Monte Carlo vs. Quasi-Monte Carlo

Simple example: $f : [0,1] \mapsto \mathbb{R}$. Estimate $\int_0^1 f(x)\,\mathrm{d}x$.

Conclusion: Generate points uniformly.

We are not assuming that the driver sequence is random, rather we think of the sequence $u_1, u_2, \ldots, \in [0,1]$ as a deterministic sequence with certain properties.

Monte Carlo vs. Quasi-Monte Carlo

Simple example: $f : [0,1] \mapsto \mathbb{R}$. Estimate $\int_0^1 f(x)\,\mathrm{d}x$.

Conclusion: Generate points uniformly.

We are not assuming that the driver sequence is random, rather we think of the sequence $u_1, u_2, \ldots, \in [0,1]$ as a deterministic sequence with certain properties.

Would such a sequence work for a MCMC algorithm?

QMC points are constructed with low discrepancy.

$$D^*_{N,d}(P) = \sup_{(a_1,\ldots,a_s)\in[0,1]^s} \left| \frac{A_N(P)}{N} - a_1 \cdots a_s \right|.$$

Low discrepancy guarantees good distribution properties, but does not take care of correlations (points highly correlated). - They fail statistical tests for randomness.

Low discrepancy guarantees good distribution properties, but does not take care of correlations (points highly correlated). - They fail statistical tests for randomness.

The property we need is called **complete uniform distribution** (CUD).

Low discrepancy guarantees good distribution properties, but does not take care of correlations (points highly correlated). - They fail statistical tests for randomness.

The property we need is called **complete uniform distribution** (CUD).
$u_1, u_2, u_3, \ldots, \in [0,1)$:

Low discrepancy guarantees good distribution properties, but does not take care of correlations (points highly correlated). - They fail statistical tests for randomness.

The property we need is called **complete uniform distribution** (CUD).

$u_1, u_2, u_3, \ldots, \in [0,1)$: Define

$\boldsymbol{u}_1^{(d)} = (u_1, \ldots, u_d), \boldsymbol{u}_2^{(d)} = (u_2, \ldots, u_{d+1}), \ldots, \in [0,1)^d$.

Low discrepancy guarantees good distribution properties, but does not take care of correlations (points highly correlated). - They fail statistical tests for randomness.

The property we need is called **complete uniform distribution** (CUD).
$u_1, u_2, u_3, \ldots, \in [0,1)$: Define
$\boldsymbol{u}_1^{(d)} = (u_1, \ldots, u_d), \boldsymbol{u}_2^{(d)} = (u_2, \ldots, u_{d+1}), \ldots, \in [0,1)^d$.

Then $(u_k)_{k \geq 1}$ CUD if

$$\forall d \geq 1 : D^*_{N,d}(\{\boldsymbol{u}_1^{(d)}, \ldots, \boldsymbol{u}_N^{(d)}\}) \to 0 \text{ as } N \to \infty$$

Low discrepancy guarantees good distribution properties, but does not take care of correlations (points highly correlated). - They fail statistical tests for randomness.

The property we need is called **complete uniform distribution** (CUD).
$u_1, u_2, u_3, \ldots, \in [0,1)$: Define
$\boldsymbol{u}_1^{(d)} = (u_1, \ldots, u_d), \boldsymbol{u}_2^{(d)} = (u_2, \ldots, u_{d+1}), \ldots, \in [0,1)^d$.

Then $(u_k)_{k \geq 1}$ CUD if

$$\forall d \geq 1 : D^*_{N,d}(\{\boldsymbol{u}_1^{(d)}, \ldots, \boldsymbol{u}_N^{(d)}\}) \to 0 \text{ as } N \to \infty$$

Van der Corput sequence not CUD, consider $d = 2$.

Roughly speaking we prove that under certain assumptions:

Roughly speaking we prove that under certain assumptions:

If driver sequence $u_1, u_2, \ldots \in [0,1)$ is CUD, then MCMC converges to correct result.

Roughly speaking we prove that under certain assumptions:

If driver sequence $u_1, u_2, \ldots \in [0,1)$ is CUD, then MCMC converges to correct result.

CUD sequences not ‘more’ random, but rather they have the right property.

Result for MCMC

Theorem

Let $\Omega \subseteq \mathbb{R}^s$ and let $\boldsymbol{x}_0 \in \Omega$, and for $i \geq 1$ let $\boldsymbol{x}_i = \phi(\boldsymbol{x}_{i-1}, \boldsymbol{u}_i)$ where ϕ is the update function of a regular MCMC with a coupling region $\mathcal{C}$. If $\boldsymbol{u}_i = (v_{d(i-1)+1}, \ldots, v_{di})$ for a CUD sequence $(v_i)_{i \geq 1}$, then $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ consistently samples π.

Result for MCMC

Theorem

Let $\Omega \subseteq \mathbb{R}^s$ and let $\boldsymbol{x}_0 \in \Omega$, and for $i \geq 1$ let $\boldsymbol{x}_i = \phi(\boldsymbol{x}_{i-1}, \boldsymbol{u}_i)$ where ϕ is the update function of a regular MCMC with a coupling region $\mathcal{C}$. If $\boldsymbol{u}_i = (v_{d(i-1)+1}, \ldots, v_{di})$ for a CUD sequence $(v_i)_{i \geq 1}$, then $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ consistently samples π.

Definition (Regular MCMC)

$\boldsymbol{x}_m = \boldsymbol{x}_m(\boldsymbol{u}_0, \ldots, \boldsymbol{u}_m)$ last point generated by Rosenblatt-Chentsov transformation. MCMC is *regular* if $f(\boldsymbol{x}_m(\boldsymbol{u}_0, \ldots, \boldsymbol{u}_m))$ is Riemann integrable on $[0,1]^{d(m+1)}$ whenever f is bounded and continuous.

Result for MCMC

Theorem

Let $\Omega \subseteq \mathbb{R}^s$ and let $\boldsymbol{x}_0 \in \Omega$, and for $i \geq 1$ let $\boldsymbol{x}_i = \phi(\boldsymbol{x}_{i-1}, \boldsymbol{u}_i)$ where ϕ is the update function of a regular MCMC with a coupling region $\mathcal{C}$. If $\boldsymbol{u}_i = (v_{d(i-1)+1}, \ldots, v_{di})$ for a CUD sequence $(v_i)_{i \geq 1}$, then $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ consistently samples π.

Definition (Regular MCMC)

$\boldsymbol{x}_m = \boldsymbol{x}_m(\boldsymbol{u}_0, \ldots, \boldsymbol{u}_m)$ last point generated by Rosenblatt-Chentsov transformation. MCMC is *regular* if $f(\boldsymbol{x}_m(\boldsymbol{u}_0, \ldots, \boldsymbol{u}_m))$ is Riemann integrable on $[0,1]^{d(m+1)}$ whenever f is bounded and continuous.

Definition (Coupling Region)

Let $\mathcal{C} \subset [0,1]^d$ have positive Jordan measure. If $u, u' \in \mathcal{C}$ implies that $\phi(\boldsymbol{x}, \boldsymbol{u}) = \phi(\boldsymbol{x}, \boldsymbol{u}')$ for all $\boldsymbol{x} \in \Omega$, then $\mathcal{C}$ is a coupling region.

Convergence

$u_1, u_2, u_3, \ldots, \in [0,1)$: Define
$\boldsymbol{u}_1^{(d)} = (u_1, \ldots, u_d), \boldsymbol{u}_2^{(d)} = (u_2, \ldots, u_{d+1}), \ldots, \in [0,1)^d$. Then
$(u_k)_{k \geq 1}$ CUD if

$$\forall d \geq 1 : D^*_{N,d}(\{\boldsymbol{u}_1^{(d)}, \ldots, \boldsymbol{u}_N^{(d)}\}) \to 0 \text{ as } N \to \infty$$

Convergence of random numbers:

$$D^*_{N,d}(P_{\text{random}}) \leq C \sqrt{\frac{\log \log N}{N}}.$$

Pseudo-random numbers similar.
Can do better than that.

Levin, Niederreiter, Shparlinski: construct $u_1, u_2, u_3, \ldots \in [0,1)$ such that

$$\forall d \geq 1 : D^*_{N,d}(\{\boldsymbol{u}_1^{(d)}, \ldots, \boldsymbol{u}_N^{(d)}\}) \leq \frac{(\log N)^{d+1}}{N} \text{ as } N \to \infty.$$

Can this convergence be passed on to MCMC algorithm?

Not if acceptance-rejection step is used.

Samplers without acceptance-rejection step: Gibbs sampler, Slice sampler;

Numerical results by Tribble & Owen shows improvement.

Numerical results by Tribble: Largest improvement for:

- Gibbs sampler
- Driver sequence is LFSR (Linear Feedback Shift Register)
- LFSR has small period length such that
- he can use the full period length.
- ‘Randomize’ the input sequence.

Thank You!