Sampling from complex probability distributions

Louis J. M. Aslett (louis.aslett@durham.ac.uk)
Department of Mathematical Sciences
Durham University

UTOPIAE Training School II
4 July 2017

UTOPIÆ Uncertainty Treatment and Optimisation in Aerospace Engineering

MARIE CURIE ACTIONS

Durham University

Motivation

Sampling from probability distributions – why?

Monte Carlo essentially avoids the quandry of choosing an accurate but intractable model versus a simple but computable one.

Sampling from probability distributions – why?

Monte Carlo essentially avoids the quandry of choosing an accurate but intractable model versus a simple but computable one.

May want to answer:

- Probabilistic questions
  - simulate physical random processes
  - concerned with some corresponding random outcome
  - may be inherent or perceived randomness
  - eg simulation of shuttle launch
- Deterministic questions
  - most often, boils down to computation of high dimensional integrals
  - use ‘experimental’ methods to answer ‘theoretical’ question

Bayesian inference (recall Georgios Karagiannis' talk)

- Data: $\underline{t} = \{t_1, \ldots, t_n\}$
- Model: $\underline{t}$ is the realisation of a random vector $\underline{T}$ having probability density $\pi_{T \mid \Psi}(\cdot \mid \psi)$, where ψ is an unknown parameter. $\pi_{T \mid \Psi}(\underline{t} \mid \cdot)$ is the *likelihood*.
- Prior: all knowledge about ψ which is not contained in $\underline{t}$ is expressed via *prior* density $\pi_{\Psi}(\psi)$.

Bayesian inference (recall Georgios Karagiannis' talk)

- Data: $\underline{t} = \{t_1, \ldots, t_n\}$
- Model: $\underline{t}$ is the realisation of a random vector $\underline{T}$ having probability density $\pi_{T \mid \Psi}(\cdot \mid \psi)$, where ψ is an unknown parameter. $\pi_{T \mid \Psi}(\underline{t} \mid \cdot)$ is the *likelihood*.
- Prior: all knowledge about ψ which is not contained in $\underline{t}$ is expressed via *prior* density $\pi_{\Psi}(\psi)$.
- Posterior: Bayes' Theorem enables us to rationally update the prior to our *posterior* belief in light of the new evidence (data).

Bayes' Theorem

$$\pi_{\Psi \mid T}(\psi \mid \underline{t}) = \frac{\pi_{T \mid \Psi}(\underline{t} \mid \psi)\, \pi_{\Psi}(\psi)}{\int_{\Omega} \pi_{T \mid \Psi}(\underline{t} \mid \psi)\, \pi_{\Psi}(d\psi)}$$

Bayesian inference (recall Georgios Karagiannis' talk)

- Data: $\underline{t} = \{t_1, \ldots, t_n\}$
- Model: $\underline{t}$ is the realisation of a random vector $\underline{T}$ having probability density $\pi_{T \mid \Psi}(\cdot \mid \psi)$, where ψ is an unknown parameter. $\pi_{T \mid \Psi}(\underline{t} \mid \cdot)$ is the *likelihood*.
- Prior: all knowledge about ψ which is not contained in $\underline{t}$ is expressed via *prior* density $\pi_{\Psi}(\psi)$.
- Posterior: Bayes' Theorem enables us to rationally update the prior to our *posterior* belief in light of the new evidence (data).

Bayes' Theorem

$$\pi_{\Psi \mid T}(\psi \mid \underline{t}) = \frac{\pi_{T \mid \Psi}(\underline{t} \mid \psi)\, \pi_{\Psi}(\psi)}{\int_{\Omega} \pi_{T \mid \Psi}(\underline{t} \mid \psi)\, \pi_{\Psi}(d\psi)} \qquad \propto \pi_{T \mid \Psi}(\underline{t} \mid \psi)\, \pi_{\Psi}(\psi)$$

Everything is about expectations ...

Recall, for a random variable $X \in \Omega$ having probability density $\pi(x)$,

$$\mathbb{E}[f(X)] := \int_{\Omega} f(x)\pi(dx) \triangleq \mu$$

Everything is about expectations ...

Recall, for a random variable $X \in \Omega$ having probability density $\pi(x)$,

$$\mathbb{E}[f(X)] := \int_{\Omega} f(x)\pi(dx) \triangleq \mu$$

Pretty much all statements of probability can be phrased in terms of expectations.

- $X \in \mathbb{R}$,

$$\mathbb{P}(X < a) = \int_{-\infty}^{a} \pi(dx) = \int_{-\infty}^{\infty} \mathbb{I}_{(-\infty,a)}(x)\pi(dx) = \mathbb{E}[\mathbb{I}_{(-\infty,a)}(X)]$$

Everything is about expectations ...

Recall, for a random variable $X \in \Omega$ having probability density $\pi(x)$,

$$\mathbb{E}[f(X)] := \int_\Omega f(x)\pi(dx) \triangleq \mu$$

Pretty much all statements of probability can be phrased in terms of expectations.

- $X \in \mathbb{R}$,

  $$\mathbb{P}(X < a) = \int_{-\infty}^{a} \pi(dx) = \int_{-\infty}^{\infty} \mathbb{I}_{(-\infty,a)}(x)\pi(dx) = \mathbb{E}[\mathbb{I}_{(-\infty,a)}(X)]$$

- $X \in \Omega$, $A \subseteq \Omega$,

  $$\mathbb{P}(X \in A) = \int_A \pi(dx) = \int_\Omega \mathbb{I}_A(x)\pi(dx) = \mathbb{E}[\mathbb{I}_A(X)]$$

  ie for statements of probability define $f(x) := \mathbb{I}_A(x)$

Bayesian inference again

- May want samples directly from the posterior;
  - marginal kernel density estimates;
  - posterior predictive simulation;
  - etc

Bayesian inference again

- May want samples directly from the posterior;
  - marginal kernel density estimates;
  - posterior predictive simulation;
  - etc
- Or may want to answer question about a probability under the posterior

$$\begin{aligned}\mathbb{P}(\psi \in A \,|\, \underline{t}) &= \int_A \pi(d\psi \,|\, \underline{t}) \\ &= \frac{\int_\Omega \mathbb{I}_A(\psi)\pi(\underline{t} \,|\, \psi)\ \pi(d\psi)}{\int_\Omega \pi(\underline{t} \,|\, \psi)\ \pi(d\psi)} \\ &= \int_\Omega \mathbb{I}_A(\psi)\pi(d\psi \,|\, \underline{t})\end{aligned}$$

So ... just do numerical integration?

Midpoint Riemann integral in 1-dim using n evaulations:

$$\int_a^b f(x)\pi(dx) = \int_a^b g(x)\,dx \approx \frac{b-a}{n}\sum_{i=1}^{n} g(x_i)$$

where

$$x_i := a + \frac{b-a}{n}\left(i - \frac{1}{2}\right)$$

So ... just do numerical integration?

Midpoint Riemann integral in 1-dim using n evaulations:

$$\int_a^b f(x)\pi(dx) = \int_a^b g(x)\,dx \approx \frac{b-a}{n}\sum_{i=1}^{n} g(x_i)$$

where

$$x_i := a + \frac{b-a}{n}\left(i - \frac{1}{2}\right)$$

It is easy to show the error is:

$$\left|\int_a^b g(x)\,dx - \frac{b-a}{n}\sum_{i=1}^{n} g(x_i)\right| \leq \frac{(b-a)^3}{24n^2}\max_{a\leq z\leq b}|f''(z)|$$

Clearly, $\frac{(b-a)^3}{24}\max_{a\leq z\leq b}|f''(z)|$ is fixed by the problem, so we achieve desired accuracy by controlling n^{-2}.

So … just do numerical integration?

- error in midpoint Riemann integral in 1-dim:

$$\mathcal{O}(n^{-2})$$

So … just do numerical integration?

- error in midpoint Riemann integral in 1-dim:

$$\mathcal{O}(n^{-2})$$

but

- error in midpoint Riemann integral in d-dim:

$$\mathcal{O}(n^{-2/d})$$

  so-called ‘curse of dimensionality’

So … just do numerical integration?

- error in midpoint Riemann integral in 1-dim:

$$\mathcal{O}(n^{-2})$$

but

- error in midpoint Riemann integral in d-dim:

$$\mathcal{O}(n^{-2/d})$$

  so-called ‘curse of dimensionality’
- error in Monte Carlo integration:

$$\mathcal{O}(n^{-1/2})$$

  ie independent of dimension

So … just do numerical integration?

- error in midpoint Riemann integral in 1-dim:

$$\mathcal{O}(n^{-2})$$

but

- error in midpoint Riemann integral in d-dim:

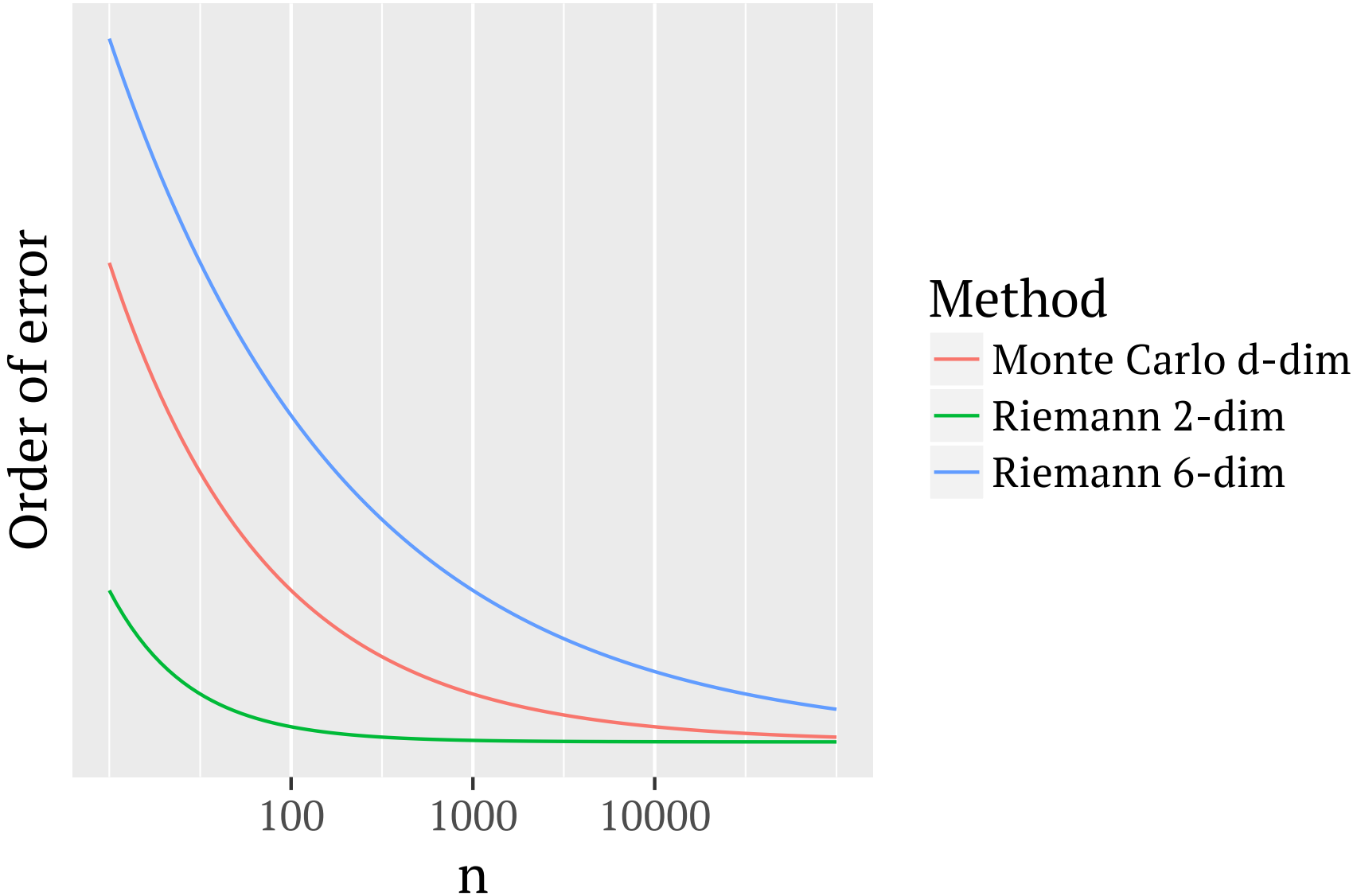
$$\mathcal{O}(n^{-2/d})$$

  so-called ‘curse of dimensionality’
- error in Monte Carlo integration:

$$\mathcal{O}(n^{-1/2})$$

  ie independent of dimension

Simpson’s improves this to $\mathcal{O}(n^{-4/d})$, but in general Bakhvalov’s Theorem bounds all possible quadriture methods by $\mathcal{O}(n^{-r/d})$ … quadriture can’t beat curse of dimensionality.



Note: this is the *order* of error, not absolute error!

Monte Carlo to the rescue?

Monte Carlo integration in d-dim using n evaulations:

$$\mu \triangleq \int f(x)\pi(dx) \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i) \triangleq \hat{\mu}$$

where $x_i \sim \pi(\cdot)$

Monte Carlo to the rescue?

Monte Carlo integration in d-dim using n evaulations:

$$\mu \triangleq \int f(x)\pi(dx) \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i) \triangleq \hat{\mu}$$

where $x_i \sim \pi(\cdot)$

The root mean square error is:

$$\sqrt{\mathbb{E}\left[\left(\int f(x)\pi(dx) - \frac{1}{n}\sum_{i=1}^{n} f(x_i)\right)^2\right]} = \frac{\sigma}{\sqrt{n}}$$

where $\sigma = \mathrm{Var}_\pi\left(f(X)\right)$.

Again, σ is (mostly) inherent to the problem, so we achieve desired accuracy by controlling $n^{-1/2}$

Monte Carlo to the rescue?

Monte Carlo integration in d-dim using n evaulations:

$$\mu \triangleq \int f(x)\pi(dx) \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i) \triangleq \hat{\mu}$$

where $x_i \sim \pi(\cdot)$

The root mean square error is:

$$\sqrt{\mathbb{E}\left[\left(\int f(x)\pi(dx) - \frac{1}{n}\sum_{i=1}^{n} f(x_i)\right)^2\right]} = \frac{\sigma}{\sqrt{n}}$$

where $\sigma = \mathrm{Var}_\pi\left(f(X)\right)$.

Again, σ is (mostly) inherent to the problem, so we achieve desired accuracy by controlling $n^{-1/2}$

Recall we can set $f(x) := \mathbb{I}_A(x)$ to compute probabilities.

Monte Carlo — the practicality

Remarkably:

- No dependence on d.
- No dependence on smoothness of integrand.
- $\frac{\sigma}{\sqrt{n}}$ can itself be directly estimated from the samples drawn.

Monte Carlo — the practicality

Remarkably:

- No dependence on d.
- No dependence on smoothness of integrand.
- $\frac{\sigma}{\sqrt{n}}$ can itself be directly estimated from the samples drawn.

but … the crux of the last slide was:

"*where* $x_i \sim \pi(\cdot)$"

Methodological research in Monte Carlo is largely preocupied with how to achieve this for complex probability distributions.

Monte Carlo — achieving desired accuracy

A simple application of Chebyshev's inequality allows us to bound how certain we are in a fully quantified way,

$$\mathbb{P}(|\hat{\mu} - \mu| \geq \varepsilon) \leq \frac{\mathbb{E}[(\hat{\mu} - \mu)^2]}{\varepsilon^2} = \frac{\sigma}{n\varepsilon^2}$$

Monte Carlo — achieving desired accuracy

A simple application of Chebyshev's inequality allows us to bound how certain we are in a fully quantified way,

$$\mathbb{P}(|\hat{\mu} - \mu| \geq \varepsilon) \leq \frac{\mathbb{E}[(\hat{\mu} - \mu)^2]}{\varepsilon^2} = \frac{\sigma}{n\varepsilon^2}$$

Or indeed, invoking the iid central limit theorem we can asymptotically state,

$$\mathbb{P}\left(\frac{\hat{\mu} - \mu}{\sigma n^{-1/2}} \leq z\right) \xrightarrow{n\to\infty} \Phi(z)$$

and so form confidence intervals for μ based on large n samples.

Simple MC

The setting

Almost all Monte Carlo procedures start from the assumption that we have available an unlimited stream of independent uniformly distributed values, typically on the interval $[0, 1]$.

We now want to study how to convert a stream

$$u_i \sim \mathrm{Unif}(0, 1)$$

into a stream

$$x_j \sim \pi(\cdot)$$

where x_j is generated by some algorithm depending on one or more u_i. In more advanced methods (see MCMC), x_j may also depend on x_{j-1} or even $x_1, \ldots, x_{j-1}$.

Inverse sampling

Let $F(x) := \mathbb{P}(X \leq x)$ be the cumulative distribution function for our target probability density function $\pi(\cdot)$.

Inverse Sampling

1. Sample $U \sim \text{Unif}(0, 1)$.
2. Set $X = F^{-1}(U)$.

Inverse sampling

Let $F(x) := \mathbb{P}(X \leq x)$ be the cumulative distribution function for our target probability density function $\pi(\cdot)$.

Inverse Sampling

1. Sample $U \sim \text{Unif}(0, 1)$.
2. Set $X = F^{-1}(U)$.

Why is $X \sim \pi(\cdot)$?

$$\begin{aligned}
\mathbb{P}(X \leq x) &= \mathbb{P}(F^{-1}(U) \leq x) \\
&= \mathbb{P}(F(F^{-1}(U)) \leq F(x)) \\
&= \mathbb{P}(U \leq F(x)) \\
&= F(x)
\end{aligned}$$

Inverse sampling

Let $F(x) := \mathbb{P}(X \leq x)$ be the cumulative distribution function for our target probability density function $\pi(\cdot)$.

Inverse Sampling

1. Sample $U \sim \text{Unif}(0, 1)$.
2. Set $X = F^{-1}(U)$.

Why is $X \sim \pi(\cdot)$?

$$\begin{aligned}
\mathbb{P}(X \leq x) &= \mathbb{P}(F^{-1}(U) \leq x) \\
&= \mathbb{P}(F(F^{-1}(U)) \leq F(x)) \\
&= \mathbb{P}(U \leq F(x)) \\
&= F(x)
\end{aligned}$$

To avoid problems with discrete distributions, we must define

$$F^{-1}(u) = \inf\{x : F(x) \geq u\}, \ \forall\, u \in [0, 1]$$

Rejection sampling

Seek a density $\tilde{\pi}(\cdot)$ we can sample from and such that

$$\pi(x) \leq c\tilde{\pi}(x) \; \forall x$$

where $c < \infty$. $\tilde{\pi}(\cdot)$ and $\pi(\cdot)$ need not be normalised.

Rejection Sampling

1. Sample $Y \sim \tilde{\pi}(\cdot)$ and $U \sim \text{Unif}(0, 1)$.
2. If $U \leq \frac{\pi(Y)}{c\tilde{\pi}(Y)}$, return Y, else return to 1.

Rejection sampling

Seek a density $\tilde{\pi}(\cdot)$ we can sample from and such that

$$\pi(x) \leq c\tilde{\pi}(x) \; \forall x$$

where $c < \infty$. $\tilde{\pi}(\cdot)$ and $\pi(\cdot)$ need not be normalised.

Rejection Sampling

1. Sample $Y \sim \tilde{\pi}(\cdot)$ and $U \sim \text{Unif}(0, 1)$.
2. If $U \leq \frac{\pi(Y)}{c\tilde{\pi}(Y)}$, return Y, else return to 1.

This is not perfectly efficient as we must iterate 1 & 2 a random number of times until acceptance, with

$$\mathbb{P}(\text{accept}) = \frac{1}{c}$$

Rejection sampling — caution, a low-d method

Consider a multi-variate Normal distribution centered at $\mathbf{0}$,

$$\pi(\mathbf{x}) = \det(2\pi\Sigma)^{-1/2} \exp\left(-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\Sigma^{-1}\mathbf{x}\right)$$

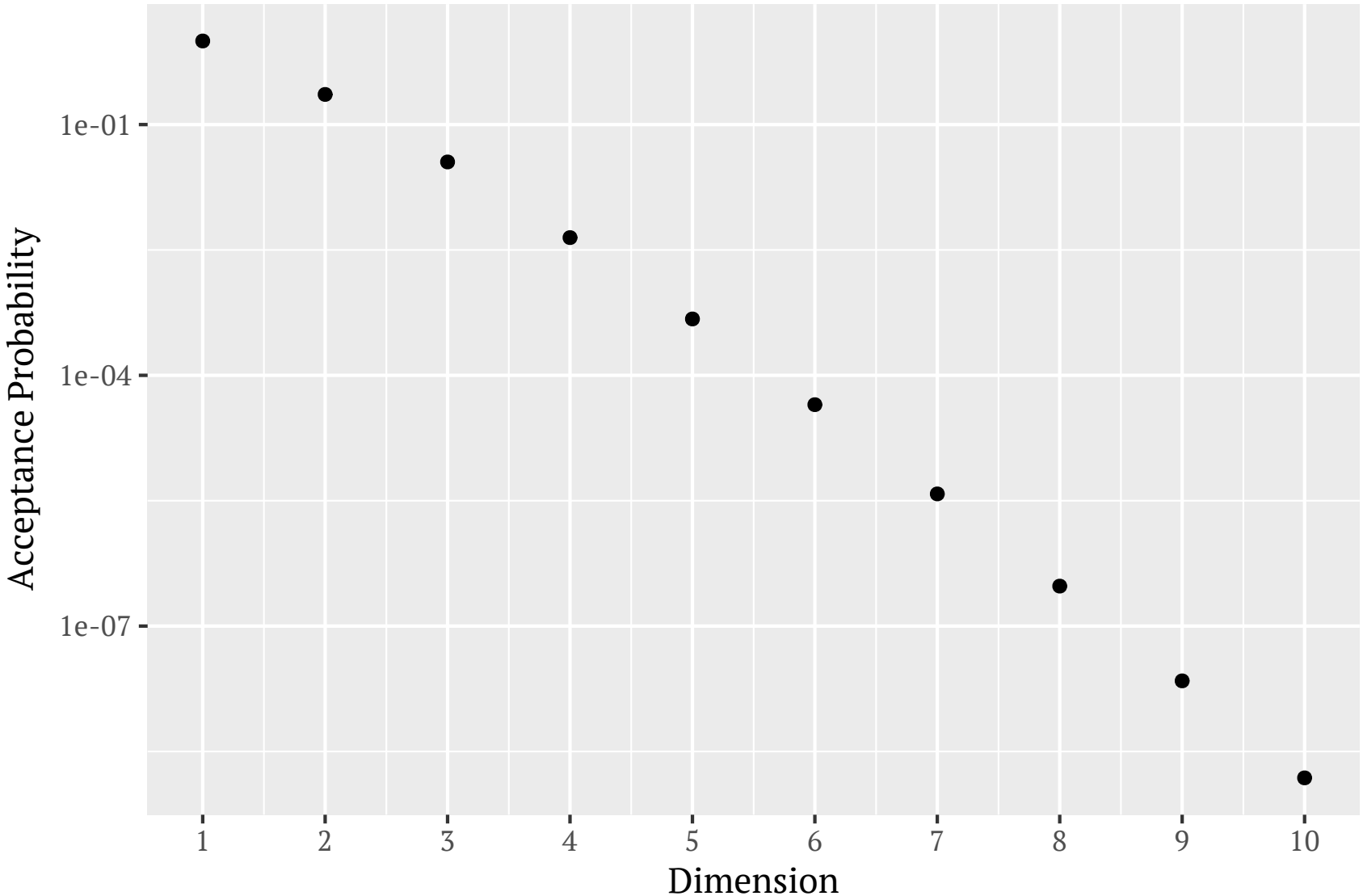
Say want to produce samples for target where

$$\Sigma = \begin{pmatrix} 1 & 0.9 & \cdots & 0.9 \\ 0.9 & 1 & \cdots & 0.9 \\ \vdots & \vdots & \ddots & \vdots \\ 0.9 & 0.9 & \cdots & 1 \end{pmatrix} = Q^{\mathrm{T}}\Lambda Q$$

using a proposal $\tilde{\pi}(\cdot)$ where $\Sigma = \sigma I$.

If $\sigma < \max\{\lambda_i\}$, $c = \infty$.

c minimal for $\sigma = \max\{\lambda_i\}$.


Acceptance Probability
1e-01
1e-04
1e-07
1
2
3
4
5
6
7
8
9
10
Dimension

Rejection sampling — demo

Shiny demo for rejection sampling with:

$$\pi(x) \propto (x-5)^2 \cos\left(x^{-1/4}\right)$$

and

$$\tilde{\pi}(x) \sim N(\mu = 4, \sigma = 3)$$

Thus,

$$c \approx 4.9 \quad \text{and} \quad \mathbb{P}(\text{accept}) \approx 0.204$$

Importance sampling (I)

If we have a probability density $\tilde{\pi}(\cdot)$ which is 'close' to $\pi(\cdot)$, then we can produce a weighted set of samples.

Importance sampling (I)

If we have a probability density $\tilde{\pi}(\cdot)$ which is 'close' to $\pi(\cdot)$, then we can produce a weighted set of samples.

Importance Sampling

1. Sample $X \sim \tilde{\pi}(\cdot)$
2. Sample is $x_i = X$, with weight $w_i = \frac{\pi(X)}{\tilde{\pi}(X)}$

Importance sampling (I)

If we have a probability density $\tilde{\pi}(\cdot)$ which is 'close' to $\pi(\cdot)$, then we can produce a weighted set of samples.

Importance Sampling

1. Sample $X \sim \tilde{\pi}(\cdot)$
2. Sample is $x_i = X$, with weight $w_i = \frac{\pi(X)}{\tilde{\pi}(X)}$

Monte Carlo estimator is slightly modified to account for weights:

$$\mu \triangleq \int f(x)\pi(dx) \approx \frac{1}{n}\sum_{i=1}^{n} w_i f(x_i) \triangleq \hat{\mu}$$

Importance sampling (I)

If we have a probability density $\tilde{\pi}(\cdot)$ which is 'close' to $\pi(\cdot)$, then we can produce a weighted set of samples.

Importance Sampling

1. Sample $X \sim \tilde{\pi}(\cdot)$
2. Sample is $x_i = X$, with weight $w_i = \frac{\pi(X)}{\tilde{\pi}(X)}$

Monte Carlo estimator is slightly modified to account for weights:

$$\mu \triangleq \int f(x)\pi(dx) \approx \frac{1}{n}\sum_{i=1}^{n} w_i f(x_i) \triangleq \hat{\mu}$$

Standard Importance Sampling Properties

$$\mathbb{E}[\hat{\mu}] = \mu, \ \mathrm{Var}(\hat{\mu}) = \frac{\sigma_{\tilde{\pi}}}{n} \quad \text{where } \sigma_{\tilde{\pi}} = \int \frac{\left(f(x)\pi(x) - \mu\tilde{\pi}(x)\right)^2}{f(x)\pi(x)}\, dx$$

Importance sampling (II)

Consequently, can show optimal proposal for importance sampling is:

$$\tilde{\pi}(x)_{\text{opt}} = \frac{|f(x)|\pi(x)}{\int_\Omega |f(x)|\pi(dx)}$$

Importance sampling (II)

Consequently, can show optimal proposal for importance sampling is:

$$\tilde{\pi}(x)_{\text{opt}} = \frac{|f(x)|\pi(x)}{\int_\Omega |f(x)|\pi(dx)}$$

Hence, importance sampling shows how to beat naïve Monte Carlo when estimating expectations of non-identity functionals — in practice, we can never compute the optimal $\tilde{\pi}(\cdot)$.

Can still provide a nice guide …

Importance sampling — unnormalised $\pi(\cdot)$

We can still perform importance sampling if $\tilde{\pi}(\cdot)$ and $\pi(\cdot)$ are only known up to a normalising constant.

Algorithm for sampling is unchanged, but the self-normalised importance sampling estimate becomes:

$$\int f(x)\pi(dx) \approx \frac{\sum_{i=1}^{n} f(x_i)w_i}{\sum_{i=1}^{n} w_i}$$

Importance sampling — unnormalised $\pi(\cdot)$

We can still perform importance sampling if $\tilde{\pi}(\cdot)$ and $\pi(\cdot)$ are only known up to a normalising constant.

Algorithm for sampling is unchanged, but the self-normalised importance sampling estimate becomes:

$$\int f(x)\pi(dx) \approx \frac{\sum_{i=1}^{n} f(x_i)w_i}{\sum_{i=1}^{n} w_i}$$

Self-normalised Importance Sampling Properties

$$\mathbb{E}[\hat{\mu}] = \mu + \frac{\mu \mathrm{Var}(W) - \mathrm{Cov}(W, Wf(X))}{n} + \mathcal{O}(n^{-2})$$

$$\mathrm{Var}(\hat{\mu}) \approx \sum_{i=1}^{n} w_i^2 (f(x_i) - \hat{\mu})^2 \quad \text{and} \quad \tilde{\pi}(x)_{\mathrm{opt}} \propto |f(x) - \mu|\pi(x)$$

Importance sampling — simple diagnostic

Equate variance of importance sampling estimate to Monte Carlo variance for a fixed sample size n_e:

$$\begin{aligned}
\mathrm{Var}\left(\frac{\sum_{i=1}^n f(x_i)w_i}{\sum_{i=1}^n w_i}\right) &= \frac{\sigma^2}{n_e} \\
\implies \frac{\mathrm{Var}\left(\sum_{i=1}^n f(x_i)w_i\right)}{\left(\sum_{i=1}^n w_i\right)^2} &= \frac{\sigma^2}{n_e} \\
\implies \frac{\sigma^2 \sum_{i=1}^n w_i^2}{\left(\sum_{i=1}^n w_i\right)^2} &= \frac{\sigma^2}{n_e} \\
\implies n_e &= \frac{n\bar{w}^2}{\overline{w^2}}
\end{aligned}$$

- balanced weights are desirable.
- small $n_e \Rightarrow$ diagnose a problem with IS
- large $n_e \not\Rightarrow$ all is ok with IS

MCMC

Markov Chain Monte Carlo

- Standard Monte Carlo methods indeed have the nice $\mathcal{O}(n^{-1/2})$ convergence *rates*
  - no dependence on dimension d
- Constant in the error term still depends on dimension!
  - no completely free lunch
- But there are methods which control the error term better than standard Monte Carlo
  - MCMC, introduced in 1953, constructs a Markov Chain whose stationary distribution is the target distribution of interest, $\pi(\cdot)$.

Markov Chains

Saw Markov Chains in an imprecise probability context yesterday morning (Gert de Cooman's talk).

Recall, a process $(X_1, X_2, \dots)$ is a continuous state space, discrete time Markov Chain if

$$\mathbb{P}(X_t \in A \,|\, X_1 = x_1, \dots, X_{t-1} = x_{t-1}) \equiv \mathbb{P}(X_t \in A \,|\, X_{t-1} = x_{t-1})$$

Markov Chains

Saw Markov Chains in an imprecise probability context yesterday morning (Gert de Cooman's talk).

Recall, a process $(X_1, X_2, \dots)$ is a continuous state space, discrete time Markov Chain if

$$\mathbb{P}(X_t \in A \,|\, X_1 = x_1, \dots, X_{t-1} = x_{t-1}) \equiv \mathbb{P}(X_t \in A \,|\, X_{t-1} = x_{t-1})$$

The transition probabilities from a current state are defined by a kernel function $K(x, \cdot)$, such that,

$$\mathbb{P}(X_t \in A \,|\, X_{t-1} = x_{t-1}) = \int_A K(x_{t-1}, dy) \triangleq K(x_{t-1}, A)$$

Markov Chains

Saw Markov Chains in an imprecise probability context yesterday morning (Gert de Cooman's talk).

Recall, a process $(X_1, X_2, \dots)$ is a continuous state space, discrete time Markov Chain if

$$\mathbb{P}(X_t \in A \,|\, X_1 = x_1, \dots, X_{t-1} = x_{t-1}) \equiv \mathbb{P}(X_t \in A \,|\, X_{t-1} = x_{t-1})$$

The transition probabilities from a current state are defined by a kernel function $K(x, \cdot)$, such that,

$$\mathbb{P}(X_t \in A \,|\, X_{t-1} = x_{t-1}) = \int_A K(x_{t-1}, dy) \triangleq K(x_{t-1}, A)$$

Under certain conditions, these chains will have a stationary distribution. We are interested in constucting Markov Chains with the stationary distribution we want to target, ie

$$\int_\Omega \pi(dx) K(x, y) = \pi(y)$$

Diving straight in ...

There is a rich and interesting theory of Markov Chains, but we'll fast-forward to the action for today.

Diving straight in ...

Metropolis-Hastings is a method to algorithmically construct $K(x, \cdot)$ such that $\pi(\cdot)$ will be stationary distribution.

Diving straight in ...

Metropolis-Hastings is a method to algorithmically construct $K(x, \cdot)$ such that $\pi(\cdot)$ will be stationary distribution.

Metropolis-Hastings

Specify a target, $\pi(\cdot)$, proposal, $q(\cdot \mid x)$, and starting point x_1.
To sample the Markov Chain, repeat:

1. Sample $x^\star \sim q(\cdot \mid x_{t-1})$
2. Compute

$$\alpha(x^\star \mid x_{t-1}) = \min\left\{1, \frac{\pi(x^\star)\, q(x_{t-1} \mid x^\star)}{\pi(x_{t-1})\, q(x^\star \mid x_{t-1})}\right\}$$

3. Sample $u \sim \text{Unif}(0, 1)$. Set,

$$x_t = \begin{cases} x^\star & \text{if } u \leq \alpha(x^\star \mid x_{t-1}) \\ x_{t-1} & \text{otherwise} \end{cases}$$

Metropolis-Hastings — common proposals

- Random-walk MH: choose some spherically symmetric distribution $g(\cdot)$ and define

$$q(x^\star \mid x) = x + \varepsilon, \text{ where } \varepsilon \sim g(\cdot)$$

  - the spherical symmetry means acceptance probability simplifies:

$$\alpha(x^\star \mid x_{t-1}) = \min\left\{1, \frac{\pi(x^\star)}{\pi(x_{t-1})}\right\}$$

  - often, $g(\cdot)$ is zero-mean multivariate Normal

Metropolis-Hastings — common proposals

- Random-walk MH: choose some spherically symmetric distribution $g(\cdot)$ and define

$$q(x^\star \mid x) = x + \varepsilon, \text{ where } \varepsilon \sim g(\cdot)$$

  - the spherical symmetry means acceptance probability simplifies:

$$\alpha(x^\star \mid x_{t-1}) = \min\left\{1, \frac{\pi(x^\star)}{\pi(x_{t-1})}\right\}$$

  - often, $g(\cdot)$ is zero-mean multivariate Normal

- Independent MH: any choice $q(x^\star \mid x) = g(x^\star)$, where the proposal does not depend on the current state.
  - generally not a good choice, easy to construct non-ergodic chains

Convergence results

We use the same estimator as standard Monte Carlo,

$$\mu \triangleq \int f(x)\pi(dx) \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i) \triangleq \hat{\mu}$$

where now x_i are MCMC draws.

Convergence results

We use the same estimator as standard Monte Carlo,

$$\mu \triangleq \int f(x)\pi(dx) \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i) \triangleq \hat{\mu}$$

where now x_i are MCMC draws.

However, we no longer have iid samples from $\pi(\cdot)$, so standard Monte Carlo convergence results do not apply. Under some mild assumptions, we can state similar results for MCMC:

$$\sqrt{n}(\hat{\mu} - \mu) \xrightarrow{n\to\infty} N(0, \sigma^2)$$

where

$$\sigma^2 = \mathrm{Var}(f(X_1)) + 2\sum_{i=2}^{\infty} \mathrm{Cov}(f(X_1), f(X_i))$$

Estimating the variance

It is hard to estimate σ^2 in the MCMC setting, but essential to be able to quantify accuracy of estimates.

- **Simple option:** always examine 'autocorrelation' plots. These will alert you to situations where the infinite sum is contributing substantially to the variance in your estimate.

- **Better option:** use methods such as batch means to estimate σ^2 from the Markov Chain output. See `mcmcse` R package for easy functions.

Choosing a proposal

- Counter-intuitively, high acceptance rates in MCMC are a *bad* thing!
    - strongly correlated draws reduce the efficiency of the estimator by inflating the variance

Choosing a proposal

- Counter-intuitively, high acceptance rates in MCMC are a *bad* thing!
    - strongly correlated draws reduce the efficiency of the estimator by inflating the variance
- But, need to move enough to explore the target
    - long range jumps which reduce correlation have very low acceptance rates

Choosing a proposal

- Counter-intuitively, high acceptance rates in MCMC are a *bad* thing!
  - strongly correlated draws reduce the efficiency of the estimator by inflating the variance
- But, need to move enough to explore the target
  - long range jumps which reduce correlation have very low acceptance rates
- Need to balance these concerns
  - a famous result shows that in the limit as $d \to \infty$, the optimal acceptance rate for a symmetric product form target density is 0.234
  - empirically this works well in lower dimensions and other targets, though for very small d should be increased (eg ≈ 0.44 in 1D)

Demo

Enough talk ...

1. Example Metropolis-Hastings sampler in R (MCMC.R)
2. MCMC convergence Shiny demo (Shiny/MCMC>R)

Adaptive MCMC

Clearly there is an issue: we may get terrible results by making a poor choice of proposal.

Adaptive MCMC

Clearly there is an issue: we may get terrible results by making a poor choice of proposal.

We can learn from the samples we have already seen to automatically improve our proposal distribution.

$$q_t(\cdot \mid x_{t-1}) = q(\cdot \mid x_{t-1}, \{x_1, \ldots, x_{t-1}\})$$

Adaptive MCMC

Clearly there is an issue: we may get terrible results by making a poor choice of proposal.

We can learn from the samples we have already seen to automatically improve our proposal distribution.

$$q_t(\cdot \mid x_{t-1}) = q(\cdot \mid x_{t-1}, \{x_1, \ldots, x_{t-1}\})$$

Warning: this breaks Markov property!

Adaptive MCMC Conditions

- **Stationarity:** $\pi(\cdot)$ must be stationary for $q_t(\cdot \mid x_{t-1})\ \forall t$
- **Diminishing adaptation:**

$$\lim_{n \to \infty} \sup_{x \in \Omega} \| K_t(\cdot \mid x) - K_{t+1}(\cdot \mid x) \| = 0$$

- **Containment:** Time to stationarity from any point in chain with adapted kernel bounded in probability.

Adaptive MCMC — Haario et al / Roberts & Rosenthal

Idea?

$$q_t(\cdot \mid x_{t-1}) \sim N\left(x_{t-1}, \frac{2.38^2}{d}\hat{\Sigma}_t\right)$$

Adaptive MCMC — Haario et al / Roberts & Rosenthal

Idea?

$$q_t(\cdot \mid x_{t-1}) \sim N\left(x_{t-1}, \frac{2.38^2}{d}\hat{\Sigma}_t\right)$$

This doesn't quite work, use

$$q_t(\cdot \mid x_{t-1}) = (1-\beta)N\left(x_{t-1}, \frac{2.38^2}{d}\hat{\Sigma}_t\right) + \beta N\left(x_{t-1}, \frac{0.1^2}{d}I\right)$$

to satisfy adaptive conditions.

Adaptive MCMC — Haario et al / Roberts & Rosenthal

Idea?

$$q_t(\cdot \,|\, x_{t-1}) \sim N\left(x_{t-1}, \frac{2.38^2}{d}\hat{\Sigma}_t\right)$$

This doesn't quite work, use

$$q_t(\cdot \,|\, x_{t-1}) = (1-\beta)N\left(x_{t-1}, \frac{2.38^2}{d}\hat{\Sigma}_t\right) + \beta N\left(x_{t-1}, \frac{0.1^2}{d}I\right)$$

to satisfy adaptive conditions.

$2.38^2/d$ is the optimal *scaling* in certain theoretical circumstances. Alternative, scale to target an acceptance rate:

$$q_t(\cdot \,|\, x_{t-1}) = (1-\beta)N\left(x_{t-1}, e^{\gamma_b}\,\hat{\Sigma}_t\right) + \beta N\left(x_{t-1}, \frac{0.1^2}{d}I\right)$$

where split into batches b of size 50, say, with

$$\gamma_b = \gamma_{b-1} + (-1)^{\mathbb{I}(\alpha_{b-1}<0.44)}\min\{0.01, n^{-1/2}\}$$

In Practice

Software

- `mcmc` R package
  - `metrop` for the kind of MCMC shown today
  - `temper` to handle multi-modality
- Stan
  - `www.mc-stan.org`
  - Hamiltonian Monte Carlo
  - several languages
- Birch
  - `www.birch-lang.org`
  - Sequential Monte Carlo
  - brand new and particularly exciting probabilistic programming language