Monte Carlo: random vectors and objects

Art B. Owen
Stanford University

Adapted from “Monte Carlo theory, methods and examples”
http://statweb.stanford.edu/~owen/mc/
Random vectors

Now we want random $\mathbf{X} = (X_1, X_2, \ldots, X_d) \in \mathbb{R}^d$.

If $X_j \sim F_j$ independent, then we’re back to the univariate case.

So the vector story is about inducing dependence.
Dependence is hard

For $d > 1$

- the correct dependence is hard to specify theoretically
- sometimes it ‘emerges’ from problem data
- our named distributions cover fewer use cases
- there can be a curse of dimension, costs like $O(e^{d \times \text{something}})$

Contrast

For $d = 1$ we could have almost any named distribution that our problem needed, or maybe build our own sampler.

For $d > 1$ we more often force our problem into a list of distributions we can do.

Special cases and tricks are prominent
(Or use MCMC or SMC.)
Sequential inversion

We want random $\mathbf{X} = (X_1, X_2, \ldots, X_d)$

Let $U_1, \ldots, U_d \overset{iid}{\sim} \mathcal{U}(0, 1)$.

Let $F_1$ be the marginal distribution of $X_1$.

$X_1 \sim F_1^{-1}(U_1)$

For $j = 2, \ldots, d$

Let $G_j(\cdot) = F_j(\cdot \mid X_1 = x_1, \ldots, X_{j-1} = x_{j-1})$

$X_j = G_j^{-1}(U_j)$

Comments

1) Exact

2) Easy if you know how

3) Ordering of variables may affect efficiency

4) Can be super hard to get all those conditional distributions
Acceptance-rejection

If \((X, Y)\) is uniformly distributed in

\[
\{(x, y) \mid 0 \leq y \leq f(x), \quad x \in \mathbb{R}^d \} \subset \mathbb{R}^{d+1}
\]

then \(X \sim f\). The geometry goes through, so the algorithm is:

1) Sample \(Y \sim g\) on \(\mathbb{R}^d\)

2) Accept iff \(f_u(Y) \leq cg_u(Y)\)

Todo list

1) Be able to sample from \(g\)

2) Be able to compute \(f_u/g_u\) (possibly unnormalized)

3) Find \(c < \infty\) where you know \(f_u \leq cg_u\)
Curse of dimension

Commonly \( c \) grows with \( d \). It can grow exponentially. Consider

\[
f = \prod_{j=1}^{d} f_j(x_j \mid x_k, \ k < j)
\]

\[
g = \prod_{j=1}^{d} g_j(x_j \mid x_k, \ k < j), \quad f_j(x_j \mid \cdots) \leq c_j g_j(x_j \mid \cdots)
\]

\[
c = \prod_{j=1}^{d} c_j
\]

If every \( c_j \geq c_0 > 1 \), then \( c \geq c_0^d \).

In a case like this we might use sequential Monte Carlo (SMC) (Chopin lectures)

If we must wait until \( X_d \) is available to accept or reject we probably face a large \( c \).
Example

We want $X \sim \mathcal{U}(B^d)$, $B^d = \{z \in \mathbb{R}^d \mid z^T z \leq 1\}$ (unit ball).

Sample $X \sim \mathcal{U}([-1, 1]^d)$ keep $X$ iff $\|X\| \leq 1$.

Round peg, square hole

\[
\begin{array}{c|c}
 d & \text{Acceptance} \\
\hline
 2 & \pi/4 \div 0.785 \\
 5 & 0.164 \\
10 & 0.00249 \\
20 & 2.46 \times 10^{-8} \\
50 & 1.54 \times 10^{-28} \\
\end{array}
\]

Generally

\[
\frac{\text{vol}(B^d)}{2^d} = \frac{\pi^{d/2}}{2^d \Gamma(1 + d/2)}
\]

Recall: $\Gamma(k) = (k - 1)!$
Mixtures

They still work.

You have to have mixing ingredients though.

So they turn $\mathbb{R}^d$ samplers into more $\mathbb{R}^d$ samplers.
Copulas

Let $\mathbf{X} \in \mathbb{R}^d$ have a continuous distribution with marginals $F_j$.

Then $\mathbf{U} = (F_1(X_1), \ldots, F_d(X_d))$ is a multivariate uniform random vector. Also called a copula.

We can take $X_j = F_j^{-1}(U_j)$ componentwise.

Sklar’s theorem

For any distribution on $\mathbb{R}^d$ there exists a copula distribution for $\mathbf{U}$ with $X_j \overset{d}{=} F_j^{-1}(U_j)$.

That doesn’t mean we can find it!

The marginals are the easy part. The copula is the hard part.
Some we *can* do

- multivariate normal
- multivariate $t$
- multinomial (multivariate binomial)
- Dirichlet (multivariate beta)
- multivariate exponential

**Puzzler**

Can we just put “multivariate” in front of any distribution name?

Sort of: but it won’t be unique. There are $\geq 12$ bivariate Gammas *(Kotz et al)*

Also “multivariate f” might not preserve meaningful properties of $f$. 
Multivariate normal

\( \mathbf{X} \sim \mathcal{N}(\mu, \Sigma), \quad \mu \in \mathbb{R}^d \) and \( \Sigma \in \mathbb{R}^{d \times d} \) positive semidefinite

\[ \mathbb{E}(\mathbf{X}) = \mu \text{ and } \text{Var}(\mathbf{X}) = \Sigma \]

Density

If \( \Sigma \) is invertible then

\[ \varphi(\mathbf{x}; \mu, \Sigma) = \frac{e^{-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)}}{(2\pi)^{d/2} |\Sigma|^{1/2}} \]

Singular distributions

Then \( \text{rank}(\Sigma) < d \) and \( \mathbf{X} \) is confined to a low dimensional flat subset of \( \mathbb{R}^d \).
\[ \mathcal{N}(\mu, \Sigma) \]

Partition: \[ \mathbf{X} = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{pmatrix} \sim \mathcal{N}\left( \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \right) \]

Key properties

1) \( A\mathbf{X} + \mathbf{b} \sim \mathcal{N}(A\mu + \mathbf{b}, A\Sigma A^T) \)

2) \( \mathbf{X}_1 \sim \mathcal{N}(\mu_1, \Sigma_{11}) \) and \( \mathbf{X}_2 \sim \mathcal{N}(\mu_2, \Sigma_{22}) \)

3) \( \mathbf{X}_1 \) indep of \( \mathbf{X}_2 \) \( \iff \) \( \Sigma_{12} = 0 \)

4) If \( \Sigma_{22} \) invertible, then distn of \( \mathbf{X}_1 \) given \( \mathbf{X}_2 = \mathbf{x}_2 \) is 
\[ \mathcal{N}\left( \mu_1 + \Sigma_{12} \Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2), \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21} \right) \]

Property 4 is our friend.
Basic $\mathcal{N}(\mu, \Sigma)$

1) Start with $Z \sim \mathcal{N}(0, I_d)$ (easy)

2) Find any $C \in \mathbb{R}^{d \times d}$ with $CC^T = \Sigma$ (below)

3) Deliver $X = \mu + CZ$

Two main choices

Cholesky: $C$ lower triangular.

Best to check $CC^T = \Sigma$. (In case you got an upper triangular $C$)

Spectral: For $\Sigma = P\Lambda P^T$ use $C = P\Lambda^{1/2}P^T$

$P$ orthogonal and $\Lambda$ diagonal

Exercise

Cholesky with $Z_j = \Phi^{-1}(U_j)$ is sequential inversion.
Gaussian

Conditional sampling is powerful. Recall $X_1 \mid X_2 = x_2$ is

$$\mathcal{N}\left(\mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (x_2 - \mu_2), \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}\right)$$

We can generate Gaussian components in any order we like.

Linear combinations

Let $T = \Theta X \in \mathbb{R}^r$ for $\Theta \in \mathbb{R}^{r \times d}$ of rank $r < d$. Then

$$\begin{pmatrix} X \\ T \end{pmatrix} = \begin{pmatrix} X \\ \Theta X \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu \\ \Theta \mu \end{pmatrix}, \begin{pmatrix} \Sigma & \Sigma \Theta^T \\ \Theta \Sigma & \Theta \Sigma \Theta^T \end{pmatrix}\right)$$

If we’ve already got $T = \Theta X$ we can fill in the rest of $X$ conditionally.

We can get $T_1 = \Theta_1 X$ then $T_2 = \Theta_2 X$.

Cost is just algebra (and careful coding).
For huge $d$

A technique from Doucet (2010)

Suppose we already chose $T = t \in \mathbb{R}^r$ where $T = \Theta X$.

Now we want to fill in the rest of $X$

We can use:

1) $X \sim \mathcal{N}(\mu, \Sigma)$

2) $X \leftarrow X + \Sigma \Theta^T (\Theta \Sigma \Theta^T)^{-1}(t - \Theta X)$

New algebra costs $O(r^3)$ not $O(d^3)$.

Still need a good $\Sigma$ sampler.
Multivariate $t$

$$X = \mu + \frac{\Sigma^{1/2} Z}{\sqrt{W/\nu}}, \quad W \sim \chi^2(\nu)$$

Elliptically symmetric contours, much heavier tails than $\mathcal{N}(\mu, \Sigma)$.

This is also a mixture of Gaussians.

- scale mixture
- continuous distribution
Multinomial data

Let $J$ be a categorical variable:

$$\mathbb{P}(J = j) = p_j \text{ for } j = 1, 2, \ldots, d$$

The “one-hot encoding” of $J = j$ is

$$Y = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{pmatrix} \in \{0, 1\}^d$$

Multinomial

$$X = \sum_{i=1}^{m} Y_i \quad \text{independent categoricals } Y_i$$

We place $m$ balls independently into $d$ bins.

Bin $j$ has probability $p_j$. 
Multinomial ctd.

\( X = (X_1, X_2, \ldots, X_d) \sim \text{Mult}(m, p) \) where \( p = (p_1, \ldots, p_d) \)

\[
P(X = x) = \frac{m!}{x_1!x_2! \cdots x_d!} \prod_{j=1}^{d} p_j^{x_j} \quad x_j \geq 0 \quad \sum_j x_j = m
\]

From the definition

\[
X \leftarrow (0, \ldots, 0) \quad \text{// length } d
\]

\[
\text{for } j = 1 \text{ to } m \text{ do}
\]

\[
J \sim p \quad \text{// i.e., } P(J = j) = p_j
\]

\[
X_j \leftarrow X_j + 1
\]

But this is slow for large \( m \).
Conditionally

We can sample them one at a time in any order we like.

Each component is binomial. Given \( X_1 = x_1 \):

\[
(X_2, \ldots, X_d) \sim \text{Mult}(m - x_1, \frac{p_2}{1-p_1}, \ldots, \frac{p_d}{1-p_1})
\]

For \( X \sim \text{Mult}(m, p) \)

\[
given m \in \mathbb{N}_0, d \in \mathbb{N} \text{ and } p = (p_1, \ldots, p_d) \in \Delta^{d-1} \\
\ell \leftarrow m, S \leftarrow 1 \\
\text{for } j = 1 \text{ to } d \text{ do} \\
\quad X_j \sim \text{Bin}(\ell, p_j / S) \\
\quad \ell \leftarrow \ell - X_j \\
\quad S \leftarrow S - p_j \\
\text{deliver } X
\]
Recursively

For any subset of bins: $u \subset \{1, 2, \ldots, d\}$

Generate $X_u \equiv \sum_{j \in u} X_j \sim \text{Bin}(m, \sum_{j \in u} p_j)$

Now you have two multinomials,

one within set $u$ and one within set $u^c$

**Fill in within set $u$**

$m \leftarrow X_u$ and $p_j \leftarrow p_j / \sum_{k \in u} p_k$

**For set $u^c$**

$m \leftarrow m - X_u$ and $p_j \leftarrow p_j / \sum_{k \in u^c} p_k$
Dirichlet

The unit simplex is

$$\Delta^{d-1} = \left\{ (x_1, \ldots, x_d) \mid x_j \geq 0, \sum_{j=1}^{d} x_j = 1 \right\}$$

A random $X \in \Delta^{d-1}$ represents a random probability vector. Useful in hierarchical models.

Density

$$D(\alpha)^{-1} \prod_{j=1}^{d} x_j^{\alpha_j-1}, \quad x \in \Delta^{d-1}, \quad D(\alpha) = \frac{\prod_{j=1}^{d} \Gamma(\alpha_j)}{\Gamma(\sum_{j=1}^{d} \alpha_j)}$$

Need $\alpha_j > 0$. If $\alpha_j = 1$ we get $U(\Delta^{d-2})$.

First $d - 1$ components

$$D(\alpha)^{-1} \prod_{j=1}^{d-1} x_j^{\alpha_j-1} \left(1 - \sum_{j=1}^{d-1} x_j\right)$$
Samples

Large $\alpha_j$ ‘attract’ points to their corner
More precisely: large $\alpha_j$ ‘repel’ points from the far side

Some Dirichlet samples
Sampling

Using some probability inequalities:

1) $Y \sim \text{Gam}(\alpha_j)$

2) $X_j = Y_j / \sum_{k=1}^d Y_k$

Marginally

This also shows that $X_j \sim \text{Beta}(\alpha_j, \sum_{k \neq j} \alpha_k)$. 
Multivariate Poisson

Take $Z_j \sim \text{Poi}(\lambda_j)$ for $j = 1, \ldots, r$ then

$$
\begin{pmatrix}
X_1 \\
X_2 \\
\vdots \\
X_d 
\end{pmatrix} = 
\begin{pmatrix}
1 & 0 & 1 & \cdots & 0 \\
0 & 1 & 0 & \cdots & 1 \\
0 & 1 & 1 & \cdots & 1
\end{pmatrix}
\begin{pmatrix}
Z_1 \\
Z_2 \\
\vdots \\
Z_r
\end{pmatrix}
$$

I.e. $X = AZ$ for $A \in \{0, 1\}^{d \times r}$

Each $X_j$ Poisson and $E(X) = A\lambda$

**Interpretation**

Event sources $Z_1, \ldots, Z_r$.

Event outcomes $X_1, \ldots, X_d$.

$A_{jk} = 1 \iff$ source $k$ affects outcome $j$.

Unfortunately: we cannot get negative dependence this way.
Copula-marginal sampling

Let $C$ be a copula. Sample $\mathbf{U} \sim C$ then $X_j = F_j^{-1}(U_j)$

Any copula we like with any margins we like.

Gaussian copula

For a correlation matrix $R \in \mathbb{R}^{d \times d}$

1) $\mathbf{Y} \sim \mathcal{N}(0, R)$

2) $\mathbf{U} \leftarrow \Phi(\mathbf{Y})$

3) $X_j \leftarrow F_j^{-1}(U_j), \quad j = 1, \ldots, d$

Also called Nataf transformation and NORTA (normal to anything).
Normal copula, Poisson margins

(a) $\rho = 0.7$

(b) $\rho = -0.7$

$\mathbb{E}(X_j) = 2$ and points jittered
Copula sampling

The Gaussian copula has some undesirable properties for insurance and finance.

A $t(\nu)$ copula is considered safer (McNeil et al., 2005)

$$Y \sim t(0, R, \nu), \quad U_j = \mathbb{P}(t(\nu) \leq Y_j), \quad X_j = F_j^{-1}(U_j)$$

Copula sampling is a hybrid with target qualitative behaviour but aesthetically problematic for some.
Geometry

Random points on

$$S^{d-1} = \{ z \in \mathbb{R}^d \mid z^T z = 1 \}$$

The standard Gaussian is spherically symmetric

$$(2\pi)^{-d/2} e^{-\frac{1}{2} z^T z}$$

Easy way to sample

1) $$Z \sim \mathcal{N}(0, I)$$
2) $$X \leftarrow Z/\|X\|$$

There are alternatives for $$d = 3$$ in graphics.

For any spherically symmetric distribution

Get $$X \sim U(S^{d-1})$$ and multiply by the desired radius.

Exercise: get $$X \sim U\{ z \in \mathbb{R}^d \mid \|z\| \leq 1 \}$$ (ball)

Box-Muller

Is this same trick in reverse to get $$Z \sim \mathcal{N}(0, I_2)$$.
Examples

Next come some sketched examples.

Time does not permit full details.

If one looks interesting, you’ll have to follow up later.
Random permutations

Uniform over $m!$ permutations of $1, \ldots, m$

\[
X \leftarrow (1, 2, \ldots, m-1, m)
\]

for $j = m, \ldots, 2$ do

\[
k \sim U\{1, \ldots, j\}
\]

swap $X_j$ and $X_k$

deliver $X$

Derangements

Exercise: Enforce $X_i \neq i$ for all $i = 1, \ldots, m$
For $K$-fold cross validation

Set up a vector with $m = K \lceil n/K \rceil$ elements

$$v = (1:K, 1:K, 1:K, \ldots, 1:K)$$

Random permutation $\pi(i)$

Group labels $G_i = v_{\pi(i)}$, $i = 1, \ldots, n$

**Fitting, tuning, validate**

Fit over 50%
tune parameters over 30%
validate on 20%
Linear permutations

To permute of $m = 2^{64}$ elements.

(Long story about min hashing)

Uniform permutation infeasible.

Suffices to permute $0, 1, \ldots, p - 1$ for prime $p > m$

Two algorithms

$$
\pi(i) = U + i \mod p \quad \text{(digital shift)}
$$

$$
\pi(i) = U + V \times i \mod p \quad \text{(random linear)}
$$

For $U \sim U\{0, 1, \ldots, p - 1\}$ and $V \sim U\{1, \ldots, p - 1\}$

NB: $V \neq 0$

These get 1 and 2 dimensional margins right (respectively).

Random linear **requires** $p$ to be prime.

These are also used in randomized quasi-Monte Carlo
Downsampling data

Given \((x_i, Y_i)\) for \(i = 1, \ldots, N\)
we want a simple random sample of \(n \ll N\)

First solution

Tag observation \(i\) with \(u_i \sim U(0, 1)\)
Keep those \(i\) with smallest \(n\) tags \(u_i\)

Better solution

Work out the distribution of ‘next item’ sampled.

Reservoir sampling

We don’t have to know \(N\) before sampling begins.
Poisson processes

Number of points in \([t, t + s) \sim \text{Poi}(\lambda \times s)\)
Non overlapping intervals are independent.

\[ T_i - T_{i-1} \sim \text{Exp}(1)/\lambda \]

Non uniform rate \(\lambda(t)\)

Let \(\Lambda(t) = \int_0^t \lambda(s) \, ds\). Then

\[ T_i = \Lambda^{-1}(\Lambda(T_{i-1}) + E_i), \quad E_i \sim \text{Exp}(1) \]

just like inversion.
Random lines

Sample via polar coordinates.

Poisson lines

Isotropic

Non-isotropic
Gaussian processes

\( X(t) \) for \( t \in \mathcal{T} \). Maybe \( \mathcal{T} = [0, \infty) \) or \( \mathcal{T} \subset \mathbb{R}^d \).

Mean \( \mu(\cdot) \) and covariance \( \Sigma(\cdot, \cdot) \).

Finite dimensional distributions

\[
\begin{pmatrix}
X(t_1) \\
X(t_2) \\
\vdots \\
X(t_m)
\end{pmatrix}
\sim \mathcal{N}
\begin{pmatrix}
\mu(t_1) \\
\mu(t_2) \\
\vdots \\
\mu(t_m)
\end{pmatrix},
\begin{pmatrix}
\Sigma(t_1, t_1) & \Sigma(t_1, t_2) & \cdots & \Sigma(t_1, t_m) \\
\Sigma(t_2, t_1) & \Sigma(t_2, t_2) & \cdots & \Sigma(t_2, t_m) \\
\vdots & \vdots & \ddots & \vdots \\
\Sigma(t_m, t_1) & \Sigma(t_m, t_2) & \cdots & \Sigma(t_m, t_m)
\end{pmatrix}
\]

Notes

We can generate in any order.
But algebra could be costly.
Easy for Brownian motion:

\[
B(t_j) = B(t_{j-1}) + \sqrt{t_j - t_{j-1}} \times \mathcal{N}(0, 1)
\]
Markov property fills in between
Matern processes

Used as generative models for functions in physics / engineering.
Supports “Bayesian numerical analysis” on expensive codes.
Stochastic differential equations

Drift $a(\cdot, \cdot)$, diffusion $b(\cdot, \cdot)$

$$dX_t = a(X_t) \, dt + b(X_t) \, dB_t,$$

Brownian motion $B_t$

Euler-Maruyama

At times $t_k = k \times \Delta$, with $Z_k \sim \mathcal{N}(0, 1)$

$$\hat{X}(t_{k+1}) = \hat{X}(t_k) + a_k \Delta + b_k \sqrt{\Delta} Z_k$$

$$a_k = a(\hat{X}(t_k)), \quad b_k = b(\hat{X}(t_k))$$

Milstein

$$\hat{X}(t_{k+1}) = \hat{X}(t_k) + a_k \Delta + b_k \sqrt{\Delta} Z_k + \frac{1}{2} b_k b'_k (Z_k^2 - 1) \Delta_k$$

$$b'_k = b'(\hat{X}(t_k))$$

Milstein’s $\hat{X}(\cdot)$ tracks $X(\cdot)$ better (strong sense).

Multilevel Monte Carlo is the best way to handle bias from $\Delta > 0$

Giles++
Dirichlet process

\( X_i \sim H(\cdot, \theta_i) \) where \( \theta_i \in \Theta \) with \( \theta_i \sim F \)

For random \( F \) centered on \( G \)

\( (F(A_1), \cdots, F(A_m)) \sim \text{Dir}(\alpha G(A_1), \ldots, \alpha G(A_m)) \)

After some algebra:

the distribution of \( \theta_{n+1} \) given \( \theta_1, \ldots, \theta_n \) is a CRP

Chinese restaurant process

Metaphor

People either start a new table
or join one with prob proportional to number seated there

Then \( \theta_{n+1} \) is either a previously seen \( \theta_i \), or a new draw from \( G \)
You get clustered \( \theta_i \) allowing for hitherto unseen clusters
Point processes


Two Spatial Point Sets

Cell centers

Finnish pines

We can mimick positive dependence via $P_i \sim \text{Poi}(\Lambda)$ for random $\Lambda$.

Negative dependence is harder.

We need MCMC lectures of Rosenthal, Roberts or SMC lectures of Chopin
Thanks

- Lecturers: Nicolas Chopin, Mark Huber, Jeffrey Rosenthal
- Guest speakers: Michael Giles, Gareth Roberts
- The London Mathematical Society: Elizabeth Fisher, Iain Stewart
- CRISM & The University of Warwick, Statistics
- Sponsors: Amazon, Google
- Partners: ISBA, MCQMC, BAYSM
- Poster: Talissa Gasser, Hidamari Design
- NSF: DMS-1407397 & DMS-1521145
- Planners: Murray Pollock, Christian Robert, Gareth Roberts
- Support: Paula Matthews, Murray Pollock, Shahin Tavakoli