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 $\boldsymbol{X} = (X_1, X_2, \ldots, X_d) \in \mathbb{R}^d$.

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 $\boldsymbol{X} = (X_1, X_2, \ldots, X_d)$

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

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 $(\boldsymbol{X}, Y)$ is uniformly distributed in

$$\{(\boldsymbol{x}, y) \mid 0 \leqslant y \leqslant f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^d\} \subset \mathbb{R}^{d+1}$$

then $\boldsymbol{X} \sim f$. The geometry goes through, so the algorithm is:

1) Sample $\boldsymbol{Y} \sim g$ on $\mathbb{R}^d$
2) Accept iff $f_u(\boldsymbol{Y}) \leqslant c g_u(\boldsymbol{Y})$

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 $f_u \leqslant c g_u$

Curse of dimension

Commonly c grows with d. It can grow exponentially. Consider

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

$$g = \prod_{j=1}^{d} g_j(x_j \mid x_k,\ k < j), \qquad f_j(x_j \mid \cdots) \leqslant c_j g_j(x_j \mid \cdots)$$

$$c = \prod_{j=1}^{d} c_j$$

If every $c_j \geqslant c_0 > 1$, then $c \geqslant c_0^d$.

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 $\boldsymbol{X} \sim \mathbf{U}(\mathbb{B}^d)$, $\quad \mathbb{B}^d = \{\boldsymbol{z} \in \mathbb{R}^d \mid \boldsymbol{z}^\mathsf{T}\boldsymbol{z} \leqslant 1\}$ (unit ball).

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

Round peg, square hole

d	Acceptance
2	$\pi/4 \doteq 0.785$
5	0.164
10	0.00249
20	2.46×10^{-8}
50	1.54×10^{-28}

Generally

$$\frac{\mathbf{vol}(\mathbb{B}^d)}{2^d} = \frac{\pi^{d/2}}{2^d\Gamma(1+d/2)}$$

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

Then $\boldsymbol{U} = (F_1(X_1), \dots, F_d(X_d))$ 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 $\boldsymbol{U}$
with $X_j \stackrel{\mathrm{d}}{=} F_j^{-1}(U_j)$.
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 $\geqslant 12$ bivariate Gammas (Kotz et al)

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

Multivariate normal

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

$\mathbb{E}(\boldsymbol{X}) = \mu$ and $\mathrm{Var}(\boldsymbol{X}) = \Sigma$

Density

If Σ is invertible then

$$\varphi(\boldsymbol{x}; \mu, \Sigma) = \frac{e^{-\frac{1}{2}(\boldsymbol{x}-\mu)^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x}-\mu)}}{(2\pi)^{d/2}|\Sigma|^{1/2}}$$

Singular distributions

Then $\mathrm{rank}(\Sigma) < d$ and $\boldsymbol{X}$ is confined to a low dimensional flat subset of $\mathbb{R}^d$.

$\mathcal{N}(\mu, \Sigma)$

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

Key properties

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

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

3) $\boldsymbol{X}_1$ indep of $\boldsymbol{X}_2 \iff \Sigma_{12} = 0$

4) If Σ_{22} invertible, then distn of $\boldsymbol{X}_1$ given $\boldsymbol{X}_2 = \boldsymbol{x}_2$ is
$$\mathcal{N}\big(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\boldsymbol{x}_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\big)$$

Property 4 is our friend.

Basic $\mathcal{N}(\mu, \Sigma)$

1) Start with $\boldsymbol{Z} \sim \mathcal{N}(0, I_d)$ (easy)
2) Find any $C \in \mathbb{R}^{d\times d}$ with $CC^\mathsf{T} = \Sigma$ (below)
3) Deliver $\boldsymbol{X} = \mu + C\boldsymbol{Z}$

Two main choices

Cholesky: C lower triangular.

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

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

P orthogonal and Λ diagonal

Execise

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

Gaussian

Conditional sampling is powerful. Recall $\boldsymbol{X}_1 \mid \boldsymbol{X}_2 = \boldsymbol{x}_2$ is

$$\mathcal{N}\left(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\boldsymbol{x}_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)$$

We can generate Gaussian components in any order we like.

Linear combinations

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

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

If we've already got $\boldsymbol{T} = \Theta\boldsymbol{X}$ we can fill in the rest of $\boldsymbol{X}$ conditionally.

We can get $\boldsymbol{T}_1 = \Theta_1\boldsymbol{X}$ then $\boldsymbol{T}_2 = \Theta_2\boldsymbol{X}$.

Cost is just algebra (and careful coding).

For huge d

A technique from Doucet (2010)

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

Now we want to fill in the rest of $\boldsymbol{X}$

We can use:

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

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

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

Still need a good Σ sampler.

Multivariate t

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

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:

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

The “one-hot encoding” of $J = j$ is

$$\boldsymbol{Y} = \begin{pmatrix} 0 & 0 & \cdots & 0 & \underbrace{1}_{\text{pos. } j} & 0 & \cdots & 0 \end{pmatrix} \in \{0, 1\}^d$$

Multinomial

$$\boldsymbol{X} = \sum_{i=1}^{m} \boldsymbol{Y}_i \quad \text{independent categoricals } \boldsymbol{Y}_i$$

We place m balls independently into d bins.

Bin j has probability p_j.

Multinomial ctd.

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

$$\mathbb{P}(\boldsymbol{X} = \boldsymbol{x}) = \frac{m!}{x_1! x_2! \cdots x_d!} \prod_{j=1}^{d} p_j^{x_j} \qquad x_j \geqslant 0 \quad \sum_j x_j = m$$

From the definition

$\boldsymbol{X} \leftarrow (0, \ldots, 0)$ // length d

for $j = 1$ to m **do**

$J \sim p$ // i.e., $\mathbb{P}(J = j) = p_j$

$X_j \leftarrow X_j + 1$

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 $X_1 = x_1$:

$$(X_2, \ldots, X_d) \sim \mathrm{Mult}\Big(m - x_1, \tfrac{p_2}{1-p_1}, \ldots, \tfrac{p_d}{1-p_1}\Big)$$

For $\boldsymbol{X} \sim \mathrm{Mult}(m, \boldsymbol{p})$

given $m \in \mathbb{N}_0, d \in \mathbb{N}$ and $\boldsymbol{p} = (p_1, \ldots, p_d) \in \Delta^{d-1}$

$\ell \leftarrow m$, $S \leftarrow 1$

for $j = 1$ to d **do**

  $X_j \sim \mathrm{Bin}(\ell, p_j/S)$

  $\ell \leftarrow \ell - X_j$

  $S \leftarrow S - p_j$

deliver $\boldsymbol{X}$

Recursively

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

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

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

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

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

Useful in hierarchical models.

Density

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

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

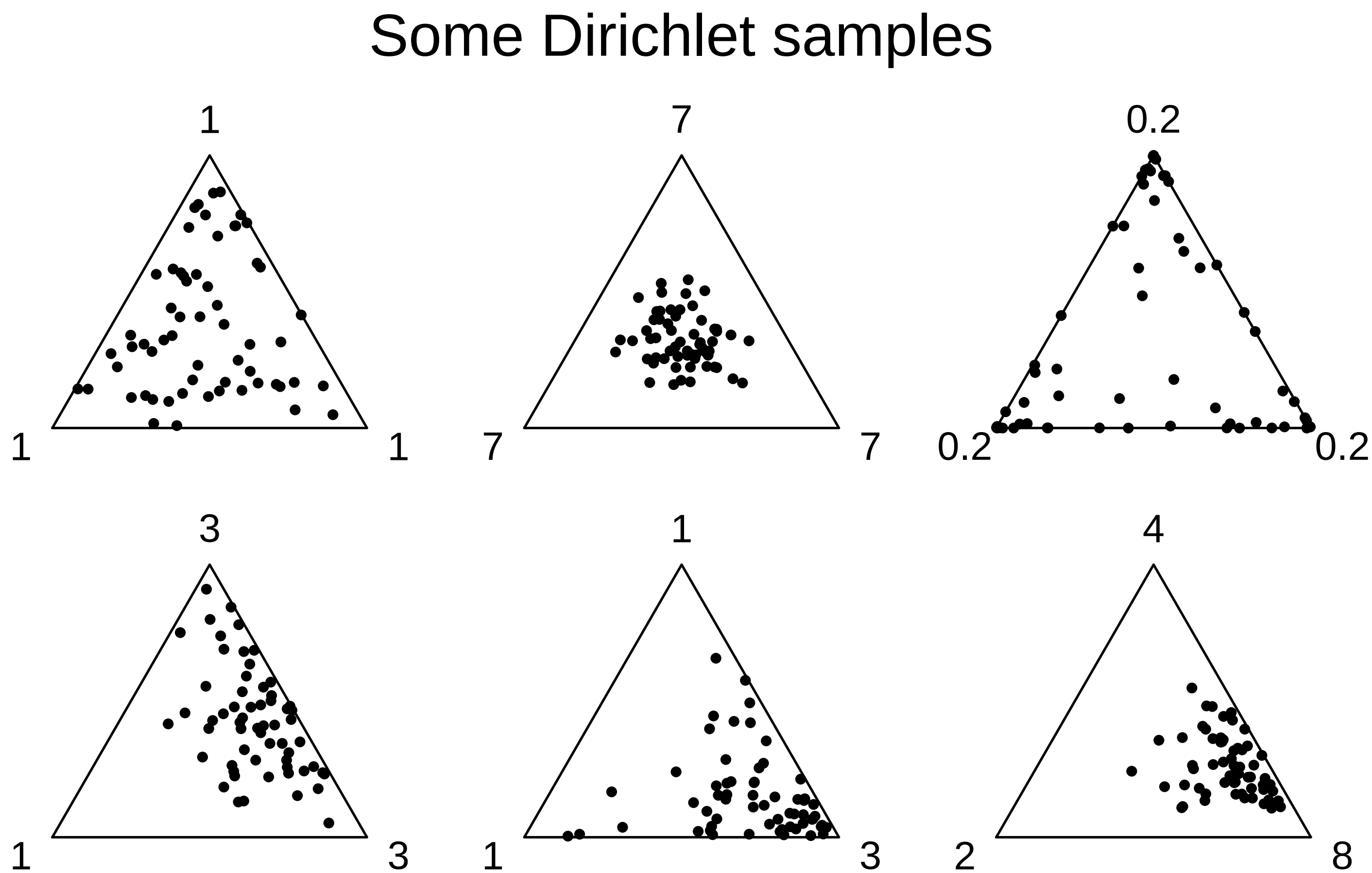
First $d-1$ components

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

Samples

Large α_j 'attract' points to their corner

More precisely: large α_j 'repel' points from the far side



Sampling

Using some probability inequalities:

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

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

Marginally

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

Multivariate Poisson

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

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

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

Each X_j Poisson and $\mathbb{E}(\boldsymbol{X}) = A\lambda$

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 $\boldsymbol{U} \sim C$ then $X_j = F_j^{-1}(U_j)$

Any copula we like with any margins we like.

Gaussian copula

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

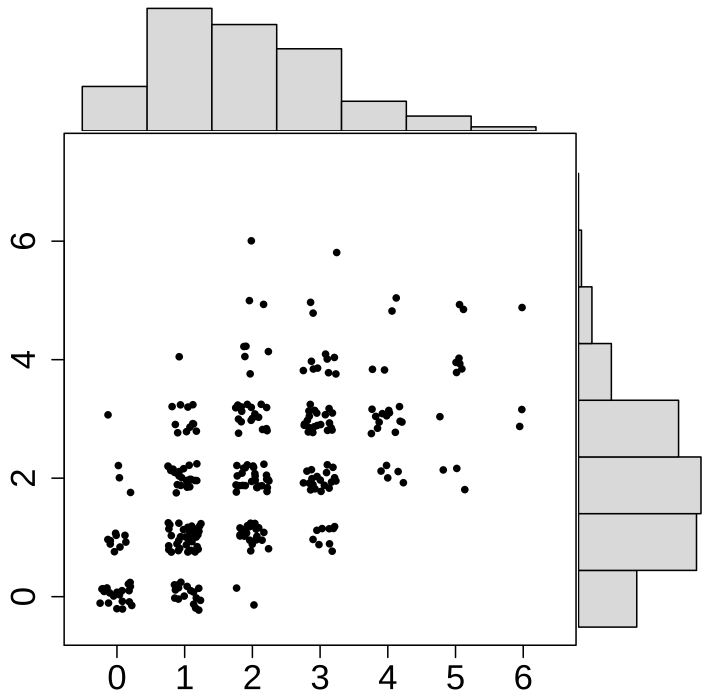
1) $\boldsymbol{Y} \sim \mathcal{N}(0, R)$

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

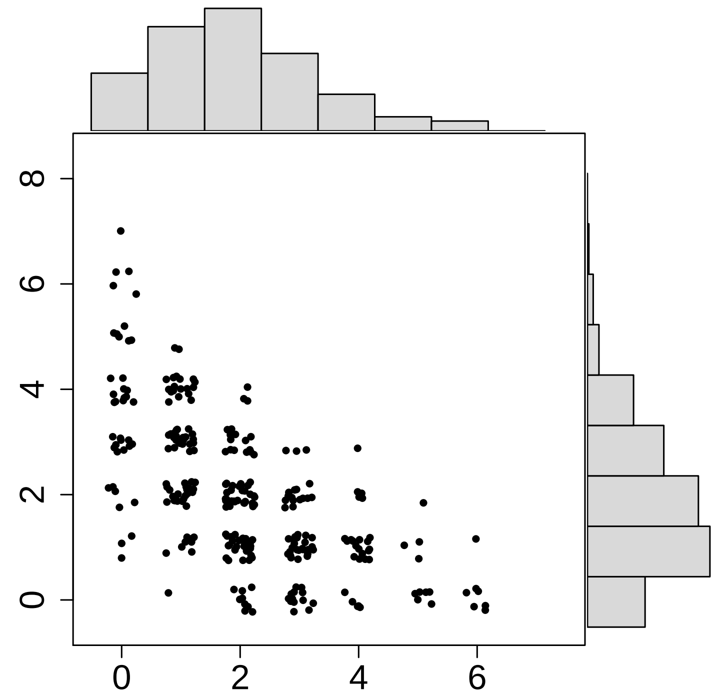
3) $X_j \leftarrow F_j^{-1}(U_j), \quad j = 1, \dots, d$

Also called Nataf transformation and NORTA (normal to anything).

Normal copula, Poisson margins



(a) $\rho = 0.7$



(b) $\rho = -0.7$

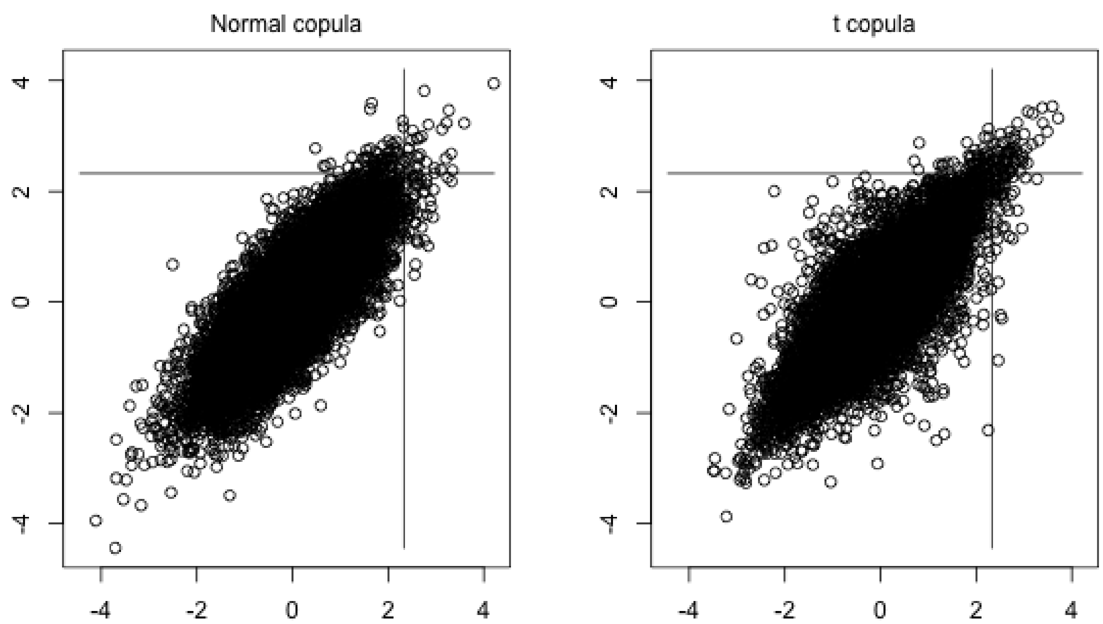
$\mathbb{E}(X_j) = 2$ and points jittered

Copula sampling

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

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

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



Copula sampling is a hybrid with target qualitative behaviour

but aesthetically problematic for some.

Geometry

Random points on

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

The standard Gaussian is spherically symmetric

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

Easy way to sample

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

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

For any spherically symmetric distribution

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

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

Box-Muller

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

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 $1, \ldots, m$

$\boldsymbol{X} \leftarrow (1, 2, \ldots, m-1, m)$

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

$k \sim \mathbf{U}\{1, \ldots, j\}$

swap X_j and X_k

deliver $\boldsymbol{X}$

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

$$\boldsymbol{v} = \big(1{:}K,\ 1{:}K,\ 1{:}K,\ \cdots\ , 1{:}K\big)$$

Random permutation $\pi(i)$

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

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 $0, 1, \dots, p-1$ for prime $p > m$

Two algorithms

$$\pi(i) = U + i \bmod p \qquad \text{(digital shift)}$$

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

For $U \sim \mathbf{U}\{0, 1, \dots, p-1\}$ and $V \sim \mathbf{U}\{1, \dots, p-1\}$
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 $(\boldsymbol{x}_i, Y_i)$ for $i = 1, \dots, N$

we want a simple random sample of $n \ll N$

First solution

Tag observation i with $u_i \sim \mathbf{U}(0,1)$

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 $[t, t+s) \sim \mathrm{Poi}(\lambda \times s)$

Non overlapping intervals are independent.

$$T_i - T_{i-1} \sim \mathrm{Exp}(1)/\lambda$$

Non uniform rate $\lambda(t)$

Let $\Lambda(t) = \int_0^t \lambda(s)\,\mathrm{d}s$. Then

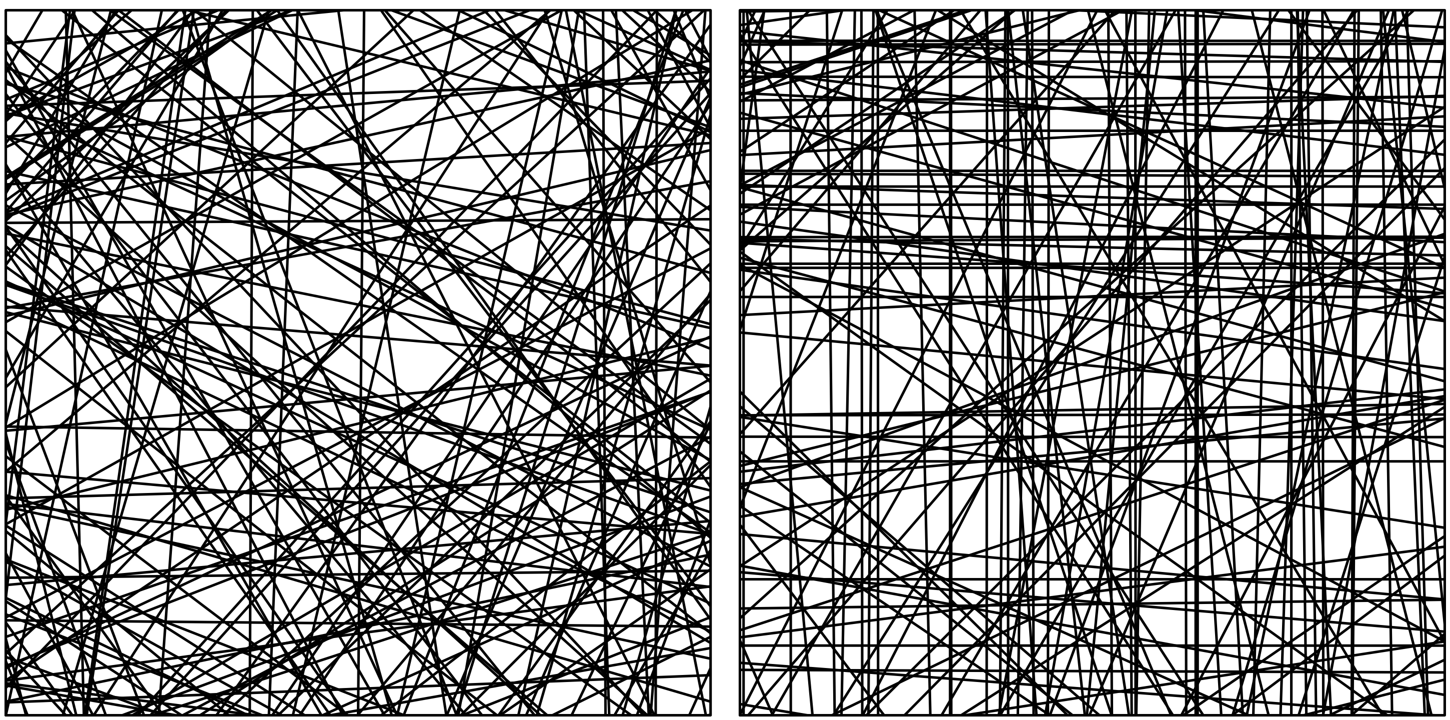
$$T_i = \Lambda^{-1}\big(\Lambda(T_{i-1}) + E_i\big), \quad E_i \sim \mathrm{Exp}(1)$$

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

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

Notes

We can generate in any order.

But algebra could be costly.

Easy for Brownian motion:

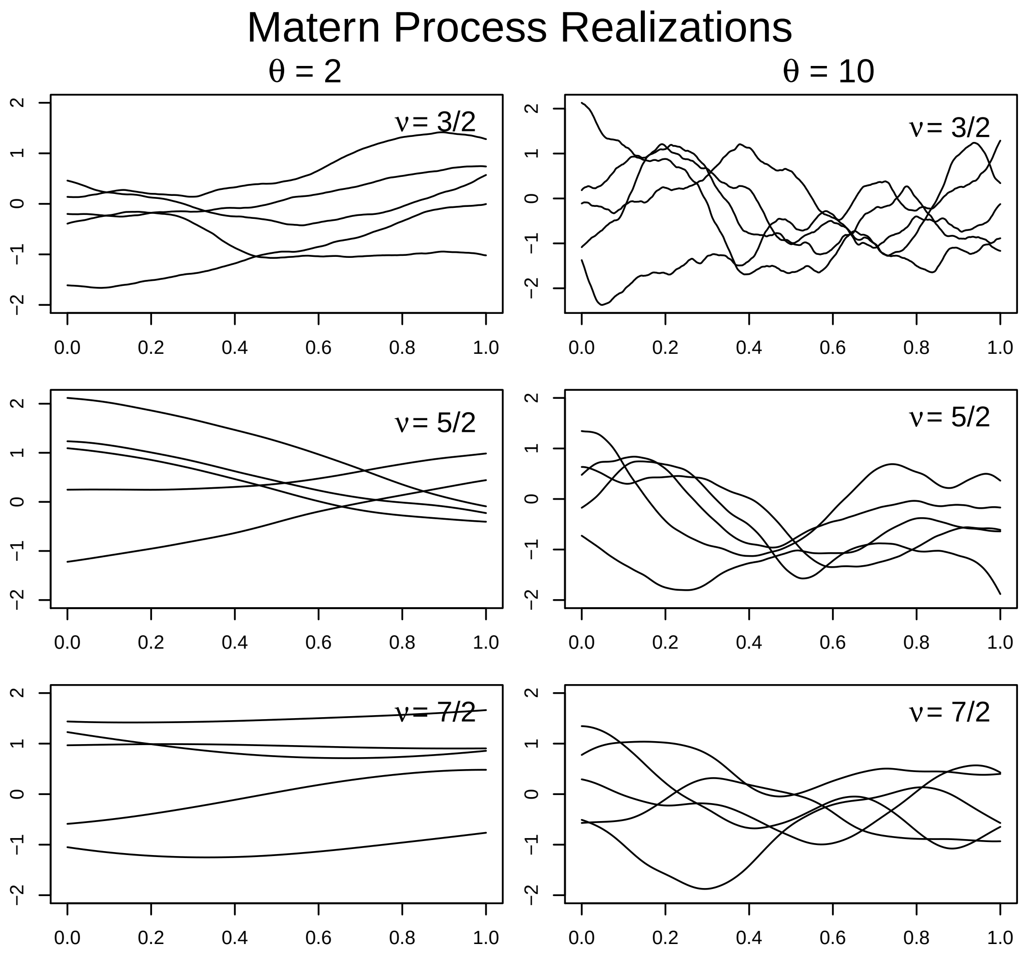
$$B(t_j) = B(t_{j-1}) + \sqrt{t_j - t_{j-1}} \times \mathcal{N}(0, 1)$$

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 $b(\cdot,\cdot)$

$$\mathrm{d}X_t = a(X_t)\,\mathrm{d}t + b(X_t)\,\mathrm{d}B_t, \quad \text{Brownian motion } B_t$$

Euler-Maruyama

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

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

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

Milstein

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

$$b_k' = b'(\widehat{X}(t_k))$$

Milstein's $\widehat{X}(\cdot)$ tracks $X(\cdot)$ better (strong sense).

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

Giles++

Dirichlet process

$\boldsymbol{X}_i \sim H(\cdot, \theta_i)$ where $\theta_i \in \Theta$ with $\theta_i \sim F$

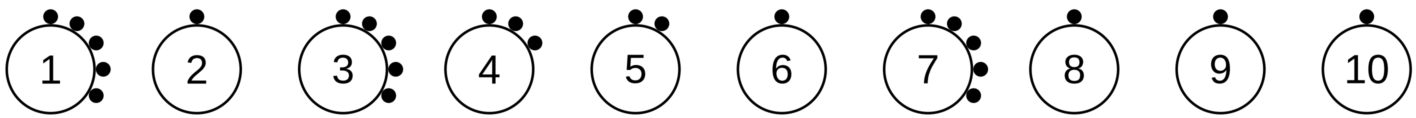
For random F centered on G

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

After some algebra:

the distribution of θ_{n+1} given $\theta_1, \ldots, \theta_n$ is a CRP

Chinese restaurant process



Metaphor

People either start a new table

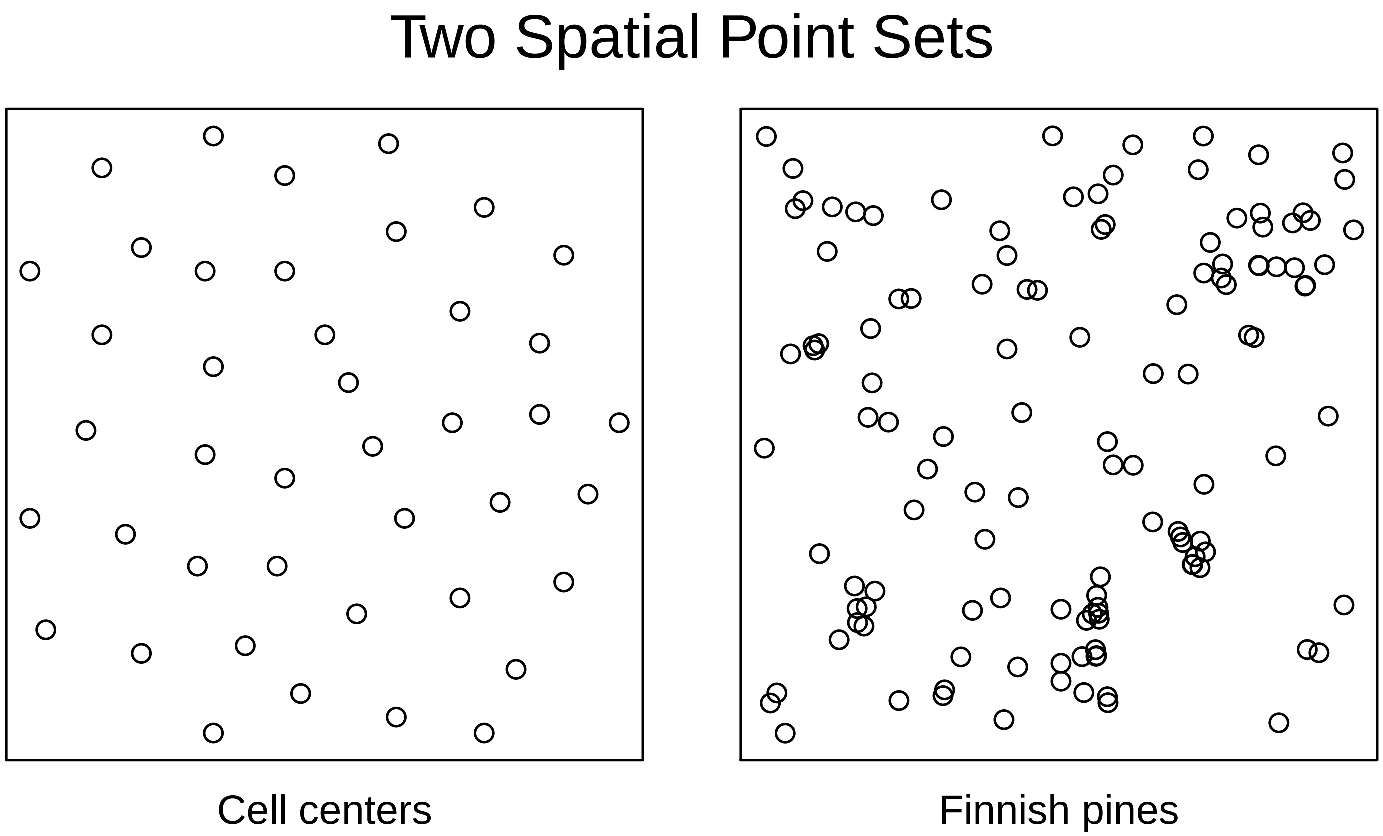
or join one with prob proportional to number seated there

Then θ_{n+1} is either a previously seen θ_i, or a new draw from G

You get clustered θ_i allowing for hitherto unseen clusters

Point processes

L: centers of insect cells Ripley (1977) R: pine trees Van Liesbout (2004)



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

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