STA3431 (Monte Carlo Methods) Lecture Notes, Fall 2021

by Professor Jeffrey Rosenthal, University of Toronto

(Last updated: November 29, 2021)

Note: These lecture notes will be posted on the STA3431 course web page after the corresponding lecture material. However, they are just rough notes, with no guarantee of completeness or accuracy. They should not be regarded as a substitute for attending and actively learning from the course lectures and supplementary readings and practice problems.

INTRODUCTION:

- Information about the course, prerequisites, etc:
  - Course web page: probability.ca/sta3431
  - Lectures: Online on Zoom, synchronous, Mondays 10:10–12:00.
  - Prerequisites: Advanced undergraduate probability/statistics/etc, plus basic computer programming (including "R"; see e.g. this page).
  - But if you already know lots about MCMC etc., then this course might not be right for you – it's an INTRODUCTION to these topics.
  - How many of you are graduate students in Statistics? Other departments? Undergrads? Auditing?
  - *** If you are not a Statistics Department graduate student, then you must REQUEST enrolment (by e-mailing the instructor).

- Theme of this course: use (pseudo)randomness on a computer to simulate, and hence estimate, important/interesting quantities.
- Example: Suppose we want to estimate $\mathbf{E}[Z^4 \cos(Z)]$, where $Z \sim \text{Normal}(0, 1)$.
  - "Classical" Monte Carlo solution: replicate a large number $z_1, \ldots, z_n$ of Normal(0,1) random variables, and let $x_i = z_i^4 \cos(z_i)$.
  - Their mean $\overline{x} \equiv \frac{1}{n} \sum_{i=1}^{n} x_i$ is an estimate of $\mathbf{E}[X] \equiv \mathbf{E}[Z^4 \cos(Z)]$.
  - We can do this in R as follows [file "RMC"]:

```
Z = rnorm(100)
X = Z^4 * cos(Z)
m = mean(X)
print(m)
```

  - Unbiased (good) ... but unstable ... but if replace "100" with "1000000" then $\overline{x}$ is consistently close to -1.213 ... good ...
  - [Note: In this course we will often use R to automatically sample from simple distributions like Normal, Uniform, Exponential, etc. But how does R do that? Discussed later!]
  - Can we quantify the variability?
  - Well, can estimate standard deviation of $\overline{x}$ by the estimated "standard error" of $\overline{x}$, which is:

$$se = \sqrt{\mathbf{Var}(\overline{x})} = \sqrt{\mathbf{Var}\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)} = \sqrt{\left(\frac{1}{n}\right)^2 \left(n\ \mathbf{var}(x)\right)}$$

$$= n^{-1/2}\sqrt{\mathbf{var}(x)} \approx n^{-1/2}\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\overline{x})^2} =: n^{-1/2}\,\mathrm{sd}(x)\,.$$

This can be computed in R [file "RMCse"] with:

```
se = sd(X) / sqrt(n)
```

- Then what is, say, a 95% confidence interval for m?
- Well, by by the Central Limit Theorem (CLT), for large n, we have $\overline{x} \approx N(m, v) \approx N(m, se^2)$.
  - (Strictly speaking, should use "t" distribution, not normal distribution ... but if n large that doesn't really matter – ignore it for now.)
  - So $\frac{m-\overline{x}}{se} \approx N(0,1)$.
  - So, $\mathbf{P}(-1.96 < \frac{m-\overline{x}}{se} < 1.96) \approx 0.95$.
  - So, $\mathbf{P}(\overline{x} - 1.96\,se < m < \overline{x} + 1.96\,se\,) \approx 0.95$.
  - i.e., approximate 95% confidence interval is

$$(\overline{x} - 1.96\,se,\ \overline{x} + 1.96\,se)\,.$$

[file "RMCci"; "cat" gives formatted output, "sep" is the separator character, and "\n" is newline]

```
cat("95% C.I.: (", m-1.96*se, ",",
                   m+1.96*se, ")\n", sep='')
```

- As an alternative, we could compute expectation as the integral

$$\int_{-\infty}^{\infty} z^4 \cos(z)\,\frac{e^{-z^2/2}}{\sqrt{2\pi}}\,dz\,.$$

Analytic? (Maybe sometimes.) Numerical integration? Yes, with:

```
f = function(z) { z^4 * cos(z) * exp(-z^2/2) /
                                 sqrt(2*3.14159) }
ival = integrate(f,-Inf,Inf)$value
print(ival)
```

Better? Worse? [file "RMCcomp": -1.213]

- [Aside: In fact, by considering it as the real part of $\mathbf{E}(Z^4 e^{iZ})$, with extra work this expectation can be computed exactly, to be $-2/\sqrt{e} \doteq -1.213061$. But not for harder examples.]
- What about higher-dimensional examples? (Can't do numerical integration!) Can we still sample?
- What if the distribution is too complicated to sample from?
  - (MCMC! Metropolis, Gibbs, etc.... Soon!)

HISTORICAL EXAMPLE – BUFFON'S NEEDLE:

- Have series of parallel lines ... line spacing w, needle length $\ell \le w$ (say $\ell = w$) ... what is prob that needle lands touching line? [file "Rbuffon"] See also e.g. https://mste.illinois.edu/activity/buffon/
- Let h be the vertical distance from the bottom end to the nearest line above, and θ be the needle's angle counter-clockwise from horizontal.
- Then $h \sim \mathrm{Uniform}[0, w]$ and $\theta \sim \mathrm{Uniform}[0, \pi]$, independent.

- Touches line iff $\ell \sin(\theta) > h$, i.e. $h < \ell \sin(\theta)$.
- So, the probability the needle touches the line is:

$$\frac{1}{\pi}\int_0^\pi \frac{1}{w}\int_0^w \mathbf{1}_{h<\ell\sin(\theta)}\, dh\, d\theta = \frac{1}{\pi}\int_0^\pi \frac{1}{w}\ell\sin(\theta)\, d\theta$$

$$= \frac{1}{\pi}\frac{1}{w}\ell\,[-\cos(\theta)]_{\theta=0}^{\theta=\pi} = \frac{1}{\pi}\frac{1}{w}\ell\,[-(-1)+(1)] = \frac{2\ell}{w\pi}\,.$$

- Hence, by LLN, if throw needle n times, of which it touches a line m times, then for n large, $m/n \approx 2\ell/w\pi$, so $\pi \approx 2n\ell/mw$.
- (e.g. if $\ell = w$, then $\pi \approx 2n/m$)
- [e.g. recuperating English Captain O.C. Fox, 1864: $\ell = 3$, $w = 4$, $n = 530$, $m = 253$, so $\pi \approx 2n\ell/mw \doteq 3.1423$.]
- But for modern simulations, use computer. How to randomise??

PSEUDORANDOM NUMBERS:

- Goal: generate an i.i.d. sequence $U_1, U_2, U_3, \ldots \sim$ Uniform$[0,1]$.
- One method: LINEAR CONGRUENTIAL GENERATOR (LCG).
  - Choose (large) positive integers m, a, and b.
  - Start with a "seed" value, x_0. (e.g., the current time in milliseconds)
  - Then, recursively, $x_n = (ax_{n-1} + b) \bmod m$, i.e. x_n = remainder when $ax_{n-1} + b$ is divided by m.
  - So, $0 \le x_n \le m - 1$.
  - Then let $U_n = x_n/m$.
  - Then $\{U_n\}$ will "seem" to be approximately i.i.d. $\sim$ Uniform$[0,1]$. [file "Rrng"; here "%%" means "remainder", and "<<–" is a global assignment so it continues outside of the function]

```
m = 2^32; a = 69069; b = 23606797  # parameters
latestval = 12345  # seed value, may be changed
nextrand = function() {
    latestval <<- (a*latestval+b) %% m
    return(latestval / m)
}
nextrand()
nextrand()
```

- Choice of m, a, and b? Many issues ...
  - Need m large (so many possible values);
  - Need a large enough to avoid similarities between U_{n-1} and U_n.
  - Need b chosen to avoid short "cycles" of numbers.
  - Want large "period", i.e. number of iterations before repeat.
  - Many statistical tests, to try to see which choices provide good randomness, avoid correlations, etc. (e.g. "diehard tests", "dieharder": www.phy.duke.edu/~rgb/General/dieharder.php)
  - One common "good" choice: $m = 2^{32}$, $a = 69,069$, $b = 23,606,797$.
- Theorem: the LCG has full period (m) if and only if both:
  (i) $\gcd(b, m) = 1$, and
  (ii) every "prime or 4" divisor of m also divides $a - 1$.

- So, if $m = 2^{32}$, then if b odd and $a-1$ is a multiple of 4 (like above), then the LCG has full period $m = 2^{32} \doteq 4.3 \times 10^9$; good.

- Aside – Many other choices, some good, some bad:
  - e.g. C programming language "glibc" uses $m = 2^{32}$, $a = 1,103,515,245$, $b = 12,345$. Pretty good.
  - "RANDU" used $m = 2^{31}$, $a = 65539 = 2^{16}+3$, $b = 0$ for many years, around the 1970s. Seemed okay. But then people noticed:
    $x_{n+2} = ax_{n+1} = a^2 x_n = (2^{16}+3)^2 x_n = (2^{32}+6(2^{16})+9)x_n$
    $\equiv (0 + 6(2^{16}+3) - 9)x_n \pmod{2^{31}} = 6x_{n+1} - 9x_n$.
    So, $x_{n+2} = 6x_{n+1} - 9x_n \bmod m$. Too much serial correlation! Bad!
  - Microsoft Excel pre-2003: had period $< 10^6$, too small ...
  - Excel 2003 instead used a floating-point "version" of LCG ... which sometimes gave negative numbers! Bad! (Fixed by 2010.)

- These numbers are <u>not</u> "really" random, just "pseudorandom" ...
  - Can cause problems! Will fail certain statistical tests!
  - Some implementations also use external randomness, e.g. temperature of computer's CPU / entropy of kernel (e.g. Linux's "urandom").
  - Or, the randomness of *quantum mechanics*, e.g. www.fourmilab.ch/hotbits (file "Rmyhotbits").
  - Or, of atmospheric noise (from lightning etc.), e.g. random.org.
  - But mostly, pseudorandom numbers work pretty well ...

- LCG's are "good enough". But other generators include (ASIDE):
  - "Multiply-with-Carry": $x_n = (ax_{n-r} + b_{n-1}) \bmod m$ where $b_n = \lfloor (ax_{n-r} + b_{n-1})/m \rfloor$.
  - 'Kiss": $y_n = (x_n + J_n + K_n) \bmod 2^{32}$, where x_n as above, and J_n and K_n are "shift register generators", given in bit form by $J_{n+1} = (I + L^{15})(I + R^{17})J_n \bmod 2^{32}$, and $K_{n+1} = (I + L^{13})(I + R^{18})K_n \bmod 2^{31}$, where L means "shift left" and R means "shift right".
  - Mersenne Twister: $x_{n+k} = x_{n+s} \oplus (x_n^{(\text{upper})} | x_{n+1}^{(\text{lower})})A$, where $1 \le s < k$ where $2^{kw-r} - 1$ is Mersenne prime, and A is $w \times w$ (e.g. 32×32) with $(w-1) \times (w-1)$ identity in upper-right, and where the matrix multiplication is done bit-wise mod 2. (Excel since 2010.)
  - And many others, too. An entire research area!
  - R's choice? See "?RNGkind". Default is Mersenne Twister.

- So, just need computer to do <u>simple arithmetic</u>. No problem, right?

LIMITATIONS OF COMPUTER ARITHMETIC:

- Consider the following computations in R:
  > 2 + 1 - 2
  > 2^10 + 1 - 2^10
  > 2^100 + 1 - 2^100

- Why??

- Question for next class: In R, for what values of n does:
  > 2^n + 1 - 2^n
  give 0 instead of 1?

- (Similarly in many other computer languages too, e.g. C (powertest.c), Java (powertest.java) ... and Python with floating numbers ... but not

Python with *integer* variables (powertest.py), because it then does dynamic memory allocation . . .)

- Also, overflow/Inf/underflow problems: 2∧10000, 2∧(−10000), 2∧10000 / 2∧10000, etc. Can cause problems in computations! (Use logs?)
- So, numerical computations are approximations, with their own errors.
- We'll usually ignore these issues, but you should BE CAREFUL!
- So how to use pseudorandomness?
  - With LCG etc, we can simulate Uniform[0,1] random variables.
  - What about other random variables?

SIMULATING OTHER DISTRIBUTIONS:

- Once we have $U_1, U_2, \ldots$ i.i.d. $\sim$ Uniform$[0,1]$ (at least approximately), how do we generate other distributions?
- With transformations, using the "change-of-variable" theorem!
- e.g. to make $X \sim$ Uniform$[L,R]$, set:
  $X = (R-L)U_1 + L$.
- e.g. to make $X \sim$ Bernoulli(p), set:

$$X = \begin{cases} 1, & U_1 \le p \\ 0, & U_1 > p \end{cases}$$

- e.g. to make $Y \sim$ Binomial(n,p), either set:
  $Y = X_1 + \ldots + X_n$ where

$$X_i \;=\; \begin{cases} 1, & U_i \le p \\ 0, & U_i > p \end{cases} ,$$

  or set:

$$Y \;=\; \max\left\{ j \;:\; \sum_{k=0}^{j-1} \binom{n}{k} p^k (1-p)^{n-k} \;\le\; U_1 \right\}$$

  (where by convention $\sum_{k=0}^{-1}(\cdots) = 0$). ("Inverse CDF method"; see below)
- More generally, to make $\mathbf{P}(Y = x_i) = p_i$ for some $x_1 < x_2 < x_3 < \ldots$, where $p_i \ge 0$ and $\sum_i p_i = 1$, set:

$$Y \;=\; \max\{x_j \,;\, \sum_{k=1}^{j-1} p_k \;\le\; U_1\}\,.$$

  (discrete version of "Inverse CDF method")

END WEEK #1

- e.g. to make $Z \sim$ Exponential(1), set:
  $Z = -\log(U_1)$.
  - Then for $x > 0$, $\mathbf{P}(Z > x) = \mathbf{P}(-\log(U_1) > x) = \mathbf{P}(\log(U_1) < -x)$ $= \mathbf{P}(U_1 < e^{-x}) = e^{-x}$. Then CDF $= 1 - e^{-x}$, and density $= e^{-x}$.
  - Then, to make $W \sim$ Exponential(λ), set:
    $W = Z/\lambda = -\log(U_1)/\lambda$. [So that W has density $\lambda e^{-\lambda x}$ for $x > 0$.]

- e.g. to make X have double-exponential (Laplace) distribution, with density $f(x) = \frac{1}{2}\, e^{-|x|}$, set

$$X = \begin{cases} -\log(U_2), & U_1 \le 1/2 \\ \log(U_2), & U_1 > 1/2 \end{cases}$$

  In R, we can do this by:

```
doubleexp = function() {
    if (runif(1) < 0.5) return( -log(runif(1)) )
    else return( log(runif(1)) )
}
```

- Suppose we want X to have density $6\, x^5 \mathbf{1}_{0<x<1}$.
  - Let $X = U_1^{1/6}$.
  - Then for $0 < x < 1$, $\mathbf{P}(X \le x) = \mathbf{P}(U_1^{1/6} \le x) = \mathbf{P}(U_1 \le x^6) = x^6$.
  - Hence, $f_X(x) = \frac{d}{dx}\Big[\mathbf{P}(X \le x)\Big] = \frac{d}{dx} x^6 = 6\, x^5$ for $0 < x < 1$.
  - More generally, for $r > 1$, if $X = U_1^{1/r}$, then $f_X(x) = r\, x^{r-1}$ for $0 < x < 1$. [CHECK!]
  - And, for $s > 0$, if $Y = U_1^{-1/s}$, then $f_Y(y) = s\, y^{-s-1}$ for $y > 1$. [CHECK!]
- What about normal dist.? Fact: If

$$X = \sqrt{2\, \log(1/U_1)}\, \cos(2\pi U_2)\,,$$

$$Y = \sqrt{2\, \log(1/U_1)}\, \sin(2\pi U_2)\,,$$

  then $X, Y \sim N(0,1)$, and X and Y are independent! ["Box-Muller transformation": Ann Math Stat 1958, 29, 610-611]
  - Proof (Aside): By multidimensional change-of-variable theorem, if $(x,y) = h(u_1,u_2)$ and $(u_1,u_2) = h^{-1}(x,y)$, then $f_{X,Y}(x,y) = f_{U_1,U_2}(h^{-1}(x,y)) \,/\, |J(h^{-1}(x,y))|$.
    Here $f_{U_1,U_2}(u_1,u_2) = 1$ for $0 < u_1, u_2 < 1$ (otherwise 0), and

$$J(u_1,u_2) = \det \begin{pmatrix} \frac{\partial x}{\partial u_1} & \frac{\partial x}{\partial u_2} \\ \frac{\partial y}{\partial u_1} & \frac{\partial y}{\partial u_2} \end{pmatrix}$$

$$= \det \begin{pmatrix} -\cos(2\pi u_2) \,/\, u_1 \sqrt{2\log(1/u_1)} & -2\pi \sin(2\pi u2) \sqrt{2\log(1/u_1)} \\ -\sin(2\pi u_2) \,/\, u_1 \sqrt{2\log(1/u_1)} & 2\pi \cos(2\pi u2) \sqrt{2\log(1/u_1)} \end{pmatrix}$$

$$= -2\pi \,/\, u_1\,.$$

    But $u_1 = e^{-(x^2+y^2)/2}$, so density of (X,Y) is

$$f_{X,Y}(x,y) = 1/|J(h^{-1}(x,y))| = 1/|-2\pi \,/\, e^{-(x^2+y^2)/2}| = e^{-(x^2+y^2)/2}/2\pi$$

$$= \Big(\frac{1}{\sqrt{2\pi}} e^{-x^2/2}\Big)\Big(\frac{1}{\sqrt{2\pi}} e^{-y^2/2}\Big)\,,$$

    i.e. $X \sim N(0,1)$ and $Y \sim N(0,1)$ are independent.
- Most general approach: the general "INVERSE CDF METHOD":
  - Suppose want $\mathbf{P}(X \le x) = F(x)$. ("CDF")
  - For $0 < t < 1$, set $F^{-1}(t) = \min\{x\,;\ F(x) \ge t\}$. ("inverse CDF")
  - Then set $X = F^{-1}(U_1)$.
  - Then $X \le x$ if and only if $U_1 \le F(x)$. [Subtle; see e.g. Rosenthal, *A First Look at Rigorous Probability Theory*, 2nd ed., Lemma 7.1.2.]

- So, $\mathbf{P}(X \le x) = \mathbf{P}(U_1 \le F(x)) = F(x)$.
- Very general, but computing $F^{-1}(t)$ is often very difficult ...

- Overall, generating (pseudo)random numbers for most "standard" one-dimensional distributions is mostly pretty easy or well-known ...
  - So, can get Monte Carlo estimates of expectations involving standard one-dimensional distributions, e.g. $\mathbf{E}[Z^4 \cos(Z)]$ where $Z \sim$ Normal$(0, 1)$.

- What about other Monte Carlo estimates?

MONTE CARLO INTEGRATION:

- How to compute an integral with Monte Carlo?
  - Re-write it as an expectation!

- EXAMPLE: Want to compute $\int_0^1 \int_0^1 g(x, y)\, dx\, dy$.
  - Regard this as $\mathbf{E}[g(X, Y)]$, where X, Y i.i.d. $\sim$ Uniform$[0, 1]$.
  - Then, similar to before, estimate $\mathbf{E}[g(X, Y)]$ by $\frac{1}{M}\sum_{i=1}^{M} g(x_i, y_i)$, where $x_i \sim$ Uniform$[0, 1]$ and $y_i \sim$ Uniform$[0, 1]$ (all independent).
  - e.g. $g(x, y) = \cos(\sqrt{xy}\,)$. [file "RMCint"]

```
g = function(x,y) { cos(sqrt(x*y)) }
M = 100000
xlist = runif(M)
ylist = runif(M)
funclist = g(xlist,ylist)
print( mean(funclist) )
print( sd(funclist) / sqrt(M) )
```

  - Get about 0.88 ± 0.003. Easy! (Mathematica gives 0.879544.)

- e.g. estimate $I = \int_0^5 \int_0^4 g(x, y)\, dy\, dx$, where $g(x, y) = \cos(\sqrt{xy}\,)$.
  - Here

$$\int_0^5 \int_0^4 g(x, y)\, dy\, dx = \int_0^5 \int_0^4 5{\cdot}4{\cdot}g(x, y)\,(1/4)dy\,(1/5)dx = \mathbf{E}[5{\cdot}4{\cdot}g(X, Y)]\,,$$

    where $X \sim$ Uniform$[0, 5]$ and $Y \sim$ Uniform$[0, 4]$.
  - So, let $X_i \sim$ Uniform$[0, 5]$, and $Y_i \sim$ Uniform$[0, 4]$ (all independent).
  - Estimate I by $\frac{1}{M}\sum_{i=1}^{M}(5 \cdot 4 \cdot g(X_i, Y_i))$. [file "RMCint2"]

```
xlist = runif(M,0,5)
ylist = runif(M,0,4)
funclist = 5 * 4 * g(xlist,ylist)
```

  - Standard error: $se = M^{-1/2}\, sd(5{\cdot}4{\cdot}g(X_1, Y_1), \ldots, 5{\cdot}4{\cdot}g(X_M, Y_M))$.
  - With $M = 10^6$, get about -4.11 ± 0.01 ... Mathematica gives -4.11692.

- e.g. estimate $\int_0^1 \int_0^\infty h(x, y)\, dy\, dx$, where $h(x, y) = e^{-y^2} \cos(\sqrt{xy}\,)$.
  - (Can't use "Uniform" expectations.)
  - Instead, write this as, say, $\int_0^1 \int_0^\infty (e^y\, h(x, y))\, e^{-y}\, dy\, dx$.
  - This is the same as $\mathbf{E}[e^Y\, h(X, Y)]$, where $X \sim$ Uniform$[0, 1]$ and $Y \sim$ Exponential(1) are independent.
  - So, estimate it by $\frac{1}{M}\sum_{i=1}^{M} e^{Y_i} h(X_i, Y_i)$, where $X_i \sim$ Uniform$[0, 1]$ and $Y_i \sim$ Exponential(1) (i.i.d.). [file "RMCint3"]

```
h = function(x,y) { exp(-y^2) * cos(sqrt(x*y)) }
xlist = runif(M)
ylist = rexp(M)
funclist = exp(ylist) * h(xlist,ylist)
```

  - With $M = 10^6$ get about $0.767 \pm 0.0004 \ldots$ Small error!
  - Mathematica: 0.767211.

- Alternatively, could write this as $\int_0^1 \int_0^\infty (\frac{1}{5}\, e^{5y}\, h(x,y))\, (5\, e^{-5y})\, dy\, dx = \mathbf{E}[\frac{1}{5} e^{5Y} h(X,Y)]$ where $X \sim$ Uniform$[0,1]$ and $Y \sim$ Exponential(5) (indep.).
  - Then, estimate it by $\frac{1}{M} \sum_{i=1}^M \frac{1}{5} e^{5y_i} h(x_i, y_i)$, where $x_i \sim$ Uniform$[0,1]$ and $y_i \sim$ Exponential(5) (i.i.d.).
  - Or more generally with any $\lambda > 0$ in place of "5". [file "RMCint4"]

```
lambda = 5
xlist = runif(M)
ylist = rexp(M, lambda)
funclist = (1/lambda) * exp(lambda*ylist) *
                                    h(xlist,ylist)
```

  - With $M = 10^6$, if $\lambda = 5$, then get about $0.767 \pm 0.0016 \ldots$ larger standard error ...
  - If replace 5 by $\lambda = 1/5$, get about $0.767 \pm 0.0015 \ldots$ about the same.

- So which choice is best?
  - Whichever one minimises the standard error!
  - ($\lambda \approx 1.5$, $se \approx 0.00025$?)

- In general, to evaluate $I \equiv \int s(y)\, dy$, could write it as $I = \int \frac{s(x)}{f(x)}\, f(x)\, dx$, where f is easily sampled from, with $f(x) > 0$ whenever $s(x) > 0$.
  - Then $I = \mathbf{E}\left(\frac{s(X)}{f(X)}\right)$, where X has density f. ("Importance Sampling")
  - So, $I \approx \frac{1}{M} \sum_{i=1}^M \frac{s(x_i)}{f(x_i)}$ where $x_i \sim f$. (***)
  - Can then do classical (iid) Monte Carlo integration, and also get standard errors, confidence intervals, etc.
  - Good if it's easier to sample from f, and/or if the function $\frac{s(x)}{f(x)}$ is less variable than h itself.

- In general, best to make $\frac{s(x)}{f(x)}$ approximately constant if possible.
  - e.g. extreme case: if $I = \int_0^\infty e^{-3x}\, dx$, then $I = \int_0^\infty (1/3)(3e^{-3x}) dx = \mathbf{E}[1/3]$ where $X \sim$ Exponential(3), so $I = 1/3$ (error $= 0$, no MC needed). [Here $s(x) = e^{-3x}$, and $f(x) = 3e^{-3x}$.]

UNNORMALISED DENSITIES:

- Suppose now that $\pi(y) = c\, g(y)$, where we know g but <u>don't</u> know c or π. ("Unnormalised density", e.g. Bayesian posterior.)
  - Obviously, $c = \frac{1}{\int g(x)\, dx}$, but this might be hard to compute.
  - Still, $I = \int h(x)\, \pi(x)\, dx = \int h(x)\, c\, g(x)\, dx = \frac{\int h(x)\, g(x)\, dx}{\int g(x)\, dx}$.
  - Using (***) above with $s(x) = h(x)\, g(x)$, we have $\int h(x)\, g(x)\, dx = \int \left(h(x)\, g(x) \,/\, f(x)\right) f(x)\, dx = \mathbf{E}[h(X)\, g(X) \,/\, f(X)]$ where $X \sim f$.

- So, $\int h(x)\,g(x)\,dx \approx \frac{1}{M}\sum_{i=1}^{M} \frac{h(x_i)\,g(x_i)}{f(x_i)}$ where $\{x_i\} \sim f$ (i.i.d.).
- And, using (***) with $s(x) = g(x)$ gives $\int g(x)\,dx \approx \frac{1}{M}\sum_{i=1}^{M} \frac{g(x_i)}{f(x_i)}$.
- So, it follows that $I \approx \dfrac{\sum_{i=1}^{M} \Big(h(x_i)\,g(x_i)\,/\,f(x_i) \Big)}{\sum_{i=1}^{M} \Big(g(x_i)\,/\,f(x_i) \Big)}$ where $\{x_i\} \sim f$ (i.i.d.).
- ("Importance Sampling": weighted average of the $h(x_i)$ values.)

- Because we are taking ratios of (unbiased) estimates, the resulting estimate is not unbiased, and its standard errors are less clear.
  - But it is still consistent, i.e. it converges to I as $M \to \infty$.

- But a direct general way to estimate the standard error of importance sampling – or any other Monte Carlo estimation method – is to repeat the same entire procedure many times, obtaining i.i.d. estimates $e_1, e_2, \ldots, e_N$, with combined estimate $\text{mean}(e_1, e_2, \ldots, e_N)$ and combined standard error $sd(e_1, e_2, \ldots, e_N)/\sqrt{N}$, just like with "RMC" etc.

- Example: compute $I \equiv \mathbf{E}(Y^2)$ where Y has density $c\,y^3 \sin(y^4) \cos(y^5)\,\mathbf{1}_{0<y<1}$, where $c > 0$ is unknown (and hard to compute).
  - Here $g(y) = y^3 \sin(y^4) \cos(y^5)\,\mathbf{1}_{0<y<1}$, and $h(y) = y^2$.
  - Let $f(y) = 6\,y^5 \mathbf{1}_{0<y<1}$.
  - [Recall: if $U \sim \text{Uniform}[0,1]$, and if $X = U^{1/6}$, then $X \sim f$.]
  - Then $I \approx \dfrac{\sum_{i=1}^{M} \big(h(x_i)\,g(x_i)\,/\,f(x_i) \big)}{\sum_{i=1}^{M} \big(g(x_i)\,/\,f(x_i) \big)} = \dfrac{\sum_{i=1}^{M} \big(\sin(x_i^4)\,\cos(x_i^5) \big)}{\sum_{i=1}^{M} \big(\sin(x_i^4)\,\cos(x_i^5)\,/\,x_i^2 \big)}$, where $\{x_i\}$ are i.i.d. $\sim f$. (file "Rimp1" ... get about 0.766 ...)

```
M = 10^6
uniflist = runif(M)
xlist = uniflist^(1/6)
numlist = sin(xlist^4) * cos(xlist^5)
denomlist = sin(xlist^4) * cos(xlist^5) / xlist^2
print( mean(numlist) / mean(denomlist) )
```

  - Or, let $f(y) = 4\,y^3 \mathbf{1}_{0<y<1}$. [So $U^{1/4} \sim f$ if $U \sim \text{Uniform}[0,1]$.]
  - Then $I \approx \dfrac{\sum_{i=1}^{M} \big(h(x_i)\,g(x_i)\,/\,f(x_i) \big)}{\sum_{i=1}^{M} \big(g(x_i)\,/\,f(x_i) \big)} = \dfrac{\sum_{i=1}^{M} \big(\sin(x_i^4)\,\cos(x_i^5)\,x_i^2 \big)}{\sum_{i=1}^{M} \big(\sin(x_i^4)\,\cos(x_i^5) \big)}$. [file "Rimp2"]

```
xlist = uniflist^(1/6)
numlist = sin(xlist^4) * cos(xlist^5)
denomlist = sin(xlist^4) * cos(xlist^5) / xlist^2
```

  - Numerical integration: 0.7661155 [file "Rimp3"].

```
numfn = function(y) { y^5 * sin(y^4) * cos(y^5) }
denomfn = function(y) { y^3 * sin(y^4) * cos(y^5) }
numval = integrate( numfn, 0, 1 )$value
denomval = integrate( denomfn, 0, 1 )$value
print(numval/denomval)
```

- With importance sampling, is it important to use the same samples $\{x_i\}$ in both numerator and denominator?
  - What if independent samples are used instead?
  - Let's try it! [file "Rimpind"]

```
uniflist = runif(M)
xlist = uniflist^(1/6)
```

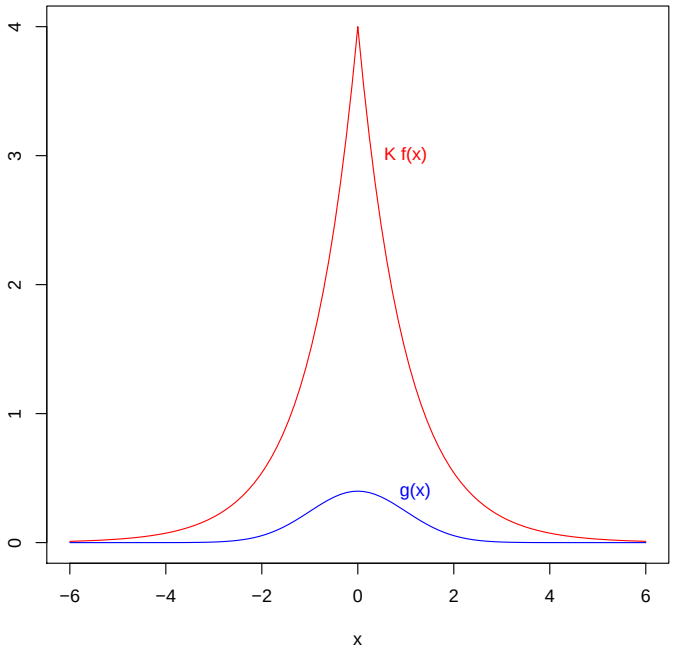
```
numlist = sin(xlist^4) * cos(xlist^5)
uniflist2 = runif(M)
xlist2 = uniflist2^(1/6)
denomlist = sin(xlist2^4) * cos(xlist2^5) / xlist2^2
print( mean(numlist) / mean(denomlist) )
```

  - Both ways work, but usually(?) the same samples work better.
  - Overall, good to use same sample $\{x_i\}$ for both numerator and denominator: easier computationally, and leads to smaller variance.
  - For example, if $h(x) = 5$ is constant, then the same samples will always give the correct answer 5, but different samples would still show lots of uncertainty.
  - (In principle, we could even use an entirely different function "f" for the numerator and the denominator ... but usually best to use the same one.)

- What other methods are available to iid sample from π?

REJECTION SAMPLER:

- Assume $\pi(x) = c\,g(x)$, with π and c unknown, g known but hard to sample from.
- Here $\int_{-\infty}^{\infty} \pi(x)\,dx = 1$, i.e. $\int_{-\infty}^{\infty} c\,g(x)\,dx = 1$, so $\int_{-\infty}^{\infty} g(x)\,dx = 1/c$.
- Want to sample $X \sim \pi$. (Then if $X_1, X_2, \ldots, X_M \sim \pi$ iid, then can estimate $\mathbf{E}_\pi(h)$ by $\frac{1}{M}\sum_{i=1}^{M} h(X_i)$, etc.)
- Find some other, easily-sampled density f, and known $K > 0$, such that $K\,f(x) \geq g(x)$ for all x. (i.e., $K\,f(x) \geq \pi(x)\,/\,c$, i.e. $c\,K\,f(x) \geq \pi(x)$)
- Sample $X \sim f$, and $U \sim \text{Uniform}[0,1]$ (indep.).
  - If $U \leq \frac{g(X)}{Kf(X)}$, then accept X (as a draw from π).
  - Otherwise, reject X and start over again.
- Does this algorithm give valid samples? Yes! (Soon ...)
- EXAMPLE: $\pi = N(0,1)$, i.e. $g(x) = \pi(x) = (2\pi)^{-1/2}\exp(-x^2/2)$.
  - Want: $\mathbf{E}_\pi(X^4)$, i.e. $h(x) = x^4$. (Should be 3.)
  - Let f be double-exponential (Laplace) distribution, i.e. $f(x) = \frac{1}{2}\,e^{-|x|}$.
- If $K = 8$, then:
  - For $|x| \leq 2$, $Kf(x) = 8\,\frac{1}{2}\,\exp(-|x|) \geq 8\,\frac{1}{2}\,\exp(-2) \geq (2\pi)^{-1/2} \geq \pi(x) = g(x)$.
  - For $|x| \geq 2$, $Kf(x) = 8\,\frac{1}{2}\,\exp(-|x|) \geq 8\,\frac{1}{2}\,\exp(-x^2/2) \geq (2\pi)^{-1/2}\exp(-x^2/2) = \pi(x) = g(x)$.
  - See graph: file "Rrejgraph"



- So, can apply rejection sampler with this f and K, to get samples, estimate of $\mathbf{E}[X]$, estimate of $\mathbf{E}[h(X)]$, estimate of $\mathbf{P}[X < -1]$, etc.
  - Try it: file "Rrej"

```
g = function(x) { dnorm(x) }
f = function(x) { 0.5 * exp(-abs(x)) }
K = 8  # constant to make K f >= g
M = 10000  # number of attempts
xlist = fulllist = rep(NA, M)
numsamples = 0
for (i in 1:M) {
    X = doubleexp()
    fulllist[i] = X  # list of *all* the values
    U = runif(1)  # for accept/reject
    if (U <- g(X) / (K * f(X)) {  # Accept X
        xlist[i] = X
        numsamples = numsamples + 1
    }
}
h = function(x) { return( x^4 ) }
cat(M, "attempts,", numsamples, "samples,",
  "estimate =", mean(h(xlist), na.rm=TRUE), "\n")
```

- Can also plot the accepted/rejected points:

```
plot(fulllist, rep(0,M), pch="|", ylim=c(-1,2))
points(xlist, rep(0.2,M), col="red", pch="|")
```

END WEEK #2

- So why does rejection sampling work?
  - First, the probability of accepting each sample is $\mathbf{P}\left(U \leq \frac{g(X)}{Kf(X)}\right)$.

- Since $0 \le \frac{g(x)}{Kf(x)} \le 1$, therefore $\mathbf{P}(U \le \frac{g(X)}{Kf(X)} \,|\, X) = \frac{g(X)}{Kf(X)}$, i.e. $\mathbf{E}\Big(\mathbf{1}_{U \le \frac{g(X)}{Kf(X)}} \,\Big|\, X\Big) = \frac{g(X)}{Kf(X)}$.
- Hence, by the Double-Expectation Formula,

$$\mathbf{P}(\text{accept}) = \mathbf{P}\Big(U \le \frac{g(X)}{Kf(X)}\Big) = \mathbf{E}\left[\mathbf{1}_{U \le \frac{g(X)}{Kf(X)}}\right]$$

$$= \mathbf{E}\left[\mathbf{E}\left(\mathbf{1}_{U \le \frac{g(X)}{Kf(X)}} \,\Big|\, X\right)\right] = \mathbf{E}\Big[\frac{g(X)}{Kf(X)}\Big]$$

$$= \int_{-\infty}^{\infty} \frac{g(x)}{Kf(x)}\, f(x)\, dx = \frac{1}{K}\int_{-\infty}^{\infty} g(x)\, dx = \frac{1}{Kc}\,.$$

- Similarly, for any $y \in \mathbf{R}$,

$$\mathbf{P}\Big(X \le y,\ U \le \frac{g(X)}{Kf(X)}\Big) = \mathbf{E}\Big[\mathbf{1}_{X \le y}\, \mathbf{1}_{U \le \frac{g(X)}{Kf(X)}}\Big]$$

$$= \mathbf{E}\Big[\mathbf{E}\Big(\mathbf{1}_{X \le y}\, \mathbf{1}_{U \le \frac{g(X)}{Kf(X)}} \Big| X\Big)\Big] = \mathbf{E}\Big[\mathbf{1}_{X \le y}\, \frac{g(X)}{Kf(X)}\Big]$$

$$= \int_{-\infty}^{y} \frac{g(x)}{Kf(x)}\, f(x)\, dx = \frac{1}{K}\int_{-\infty}^{y} g(x)\, dx\,.$$

- Then, conditional on accepting, the CDF of X is:

$$\mathbf{P}\Big(X \le y \,\Big|\, U \le \frac{g(X)}{Kf(X)}\Big) = \frac{\mathbf{P}\Big(X \le y,\ U \le \frac{g(X)}{Kf(X)}\Big)}{\mathbf{P}\Big(U \le \frac{g(X)}{Kf(X)}\Big)}$$

$$= \mathbf{P}\Big(X \le y \,\Big|\, U \le \frac{g(X)}{Kf(X)}\Big) = \frac{\frac{1}{K}\int_{-\infty}^{y} g(x)\, dx}{\frac{1}{K}\int_{-\infty}^{\infty} g(x)\, dx}$$

$$= \frac{\int_{-\infty}^{y} g(x)\, dx}{1/c} = c\int_{-\infty}^{y} g(x)\, dx = \int_{-\infty}^{y} \pi(x)\, dx\,.$$

- So, conditional on accepting, $X \sim \pi$. Good! iid!

- But $\mathbf{P}(\text{accept}) = \frac{1}{Kc}$. Good?
  - Means that with M attempts, get about M/Kc iid samples.
  - ("Rrej" example: $c = 1$, $K = 8$, $M = 10,000$, so get about $M/8 = 1250$ samples.)
  - This acceptance probability might be very small. If so, then we will get very few samples – bad.
  - Only depends on K and c, not directly on f. So, since c is fixed, try to choose f to minimise the value of K.
  - Extreme case: $f(x) = \pi(x)$, so $g(x) = \pi(x)/c = f(x)/c$, and can take $K = 1/c$, whence $\mathbf{P}(\text{accept}) = 1$, iid sampling: optimal.

AUXILIARY VARIABLE APPROACH:

- (related: "slice sampler")
- Suppose $\pi(x) = c\, g(x)$, and (X, Y) chosen uniformly under graph of g.
  - i.e., $(X, Y) \sim \text{Uniform}\{(x, y) \in \mathbf{R}^2 : 0 \le y \le g(x)\}$.
  - Then $X \sim \pi$, i.e. we have sampled from π.

- Why? Well, for $a < b$,

$$\mathbf{P}(a < X < b) = \frac{\text{area with } a < x < b}{\text{total area}} = \frac{\int_a^b g(x)\,dx}{\int_{-\infty}^{\infty} g(x)\,dx} = \int_a^b \pi(x)\,dx\,.$$

- So, if repeat, get i.i.d. samples from π, can estimate $\mathbf{E}_\pi(h)$ etc.

- Auxiliary Variable rejection sampler:
  - If support of g contained in $[L, R]$, and $|g(x)| \leq K$, then can first sample $(X, Y) \sim \text{Uniform}([L, R] \times [0, K])$, then reject if $Y > g(X)$, otherwise accept as sample with $(X, Y) \sim \text{Uniform}\{(x, y) : 0 \leq y \leq g(x)\}$, hence $X \sim \pi$.

- Example: $g(y) = y^3 \sin(y^4) \cos(y^5)\, \mathbf{1}_{0<y<1}$.
  - Then $L = 0$, $R = 1$, $K = 1$.
  - So, sample $X, Y \sim \text{Uniform}[0, 1]$, then keep X iff $Y \leq g(X)$.
  - If $h(y) = y^2$, could compute e.g. $\mathbf{E}_\pi(h)$ as the mean of the squares of the accepted samples. [file "Raux"]

```
g = function(y) { y^3 * sin(y^4) * cos(y^5) };
M = 10^4; xlist = runif(M); ylist = runif(M)
pilist = xlist[ylist <= g(xlist)];
len = length(pilist);
cat(len, "samples out of", M, "attempts.\n");
hlist = pilist^2;
cat("E(h) estimate = ", mean(hlist), "\n");
cat("E(h) se = ", sd(hlist)/sqrt(len), "\n");
```

- Again, can plot the accepted/rejected points and x values:

```
plot(xlist,ylist, pch=".")
points(xlist[ylist<=g(xlist)],ylist[ylist<=g(xlist)],
                              col="green", pch=".")
points(xlist[ylist<=g(xlist)], rep(0, len),
                              col="blue", pch="|")
```

- GENERAL NOTE: The above algorithms all work in discrete cases too.
  - Can let π, f, etc. be "probability functions", i.e. probability densities with respect to counting measure.
  - Then the algorithms proceed exactly as before.

DIGRESSION – QUEUEING THEORY:

- Consider a long line (queue) of customers.
  - Let $Q(t)$ = number of people in queue at time $t \geq 0$.

- Suppose service times $\sim$ Exponential(μ) [mean $1/\mu$], and interarrival times $\sim$ Exponential(λ) ("M/M/1 queue"), so $\{Q(t)\}$ Markovian. Then well known [e.g. STA447/2006]:
  - If $\mu \leq \lambda$, then $Q(t) \to \infty$ as $t \to \infty$.
  - If $\mu > \lambda$, then $Q(t)$ converges in distribution as $t \to \infty$:
  - $\mathbf{P}(Q(t) = i) \to (1 - \frac{\lambda}{\mu})(\frac{\lambda}{\mu})^i$, for $i = 0, 1, 2, \ldots$.
  - Check by simulation! (e.g. $\mu = 3$, $\lambda = 2$, $t = 1000$) [file "Rqueue"]

- Now suppose instead that service times $\sim$ Uniform$[0,1]$, and interarrival times have distribution of $|Z|$ where $Z \sim$ Normal$(0,1)$. Limits not easily computed. Now what?
  - Simulate it! [file "Rqueue2"]
- Or, to make the means the same as the first example, suppose service times $\sim$ Uniform$[0,2/3]$, and interarrival times have distribution of $Z^2/2$ where $Z \sim$ Normal$(0,1)$. Now what? [file "Rqueue3"]

DIGRESSION – MONTE CARLO IN FINANCE:

- X_t = stock price at time t
- Assume that $X_0 = a > 0$, and $dX_t = bX_t dt + \sigma X_t dB_t$, where $\{B_t\}$ is Brownian motion. ("diffusion")
  - i.e., for small $h > 0$,

$$(X_{t+h} - X_t \,|\, X_t) \;\approx\; bX_t(t+h-t) + \sigma X_t(B_{t+h} - B_t) \;\sim\; bX_t h + \sigma X_t \, N(0,h)\,,$$

    so

$$(X_{t+h} \,|\, X_t) \;\sim\; N\left(X_t + bX_t h,\; \sigma^2 (X_t)^2 h\right)\,. \qquad (*)$$

- A "European call option" is the option to purchase one share of the stock at a fixed time $T > 0$ for a fixed price $q > 0$.
- Question: what is a fair price for this option?
  - At time T, its value is $\max(0,\; X_T - q)$.
  - So, at time 0, its value is $e^{-rT} \max(0,\; X_T - q)$, where r is the "risk-free interest rate".
  - But at time 0, X_T is unknown! So, what is fair price??
- FACT: the fair price is equal to $\mathbf{E}(e^{-rT} \max(0,\; X_T - q))$, but only after replacing b by r.
  - (Proof: transform to risk-neutral martingale measure ...)
  - Intuition: if b very large, might as well just buy stock itself.
- If σ and r constant, then there's a formula ("Black-Scholes eqn") for this price, in terms of Φ = cdf of $N(0,1)$:

$$a\,\Phi\left(\frac{1}{\sigma\sqrt{T}}\Big(\log(a/q) + T(r + \frac{1}{2}\sigma^2)\Big)\right) - qe^{-rT}\Phi\left(\frac{1}{\sigma\sqrt{T}}\Big(\log(a/q) + T(r - \frac{1}{2}\sigma^2)\Big)\right)$$

- But we can also estimate it through (iid) Monte Carlo!
  - Use $(*)$ above (for fixed small $h > 0$, e.g. $h = 0.05$) to generate samples from the difusion.
  - Any one run is highly variable. (file "RBS", with $M = 1$)
  - But many runs give good estimate. (file "RBS", with $M = 1000$)
- An "Asian call option" is similar, but with X_T replaced by $\overline{X}_{k,t} \equiv \frac{1}{k}\sum_{i=1}^{k} X_{iT/k}$, for some fixed positive integer k (e.g., $k = 8$).
  - Above "FACT" still holds (again with X_T replaced by $\overline{X}_{k,t}$).
  - Now formulas not so simple ... but can still simulate! [file "RAO"]
- So, can iid / importance / rejection / auxiliary sampling solve ALL of our problems? No!

- Many challenging cases arise, e.g. from Bayesian statistics (later).
- Some are high-dimensional, and the above algorithms fail.
- Alternative algorithms: MCMC!

END WEEK #3

*** MARKOV CHAIN MONTE CARLO (MCMC) ***:

- Suppose have complicated, high-dimensional density $\pi = c\,g$.
- Want samples $X_1, X_2, \ldots \sim \pi$. (Then can do Monte Carlo.) Difficult!
- Define a Markov chain (dependent random process: STA2006) $X_0, X_1, X_2, \ldots$ in such a way that for large enough n, $X_n \approx \pi$.
- Then can estimate $\mathbf{E}_\pi(h) \equiv \int h(x)\,\pi(x)\,dx$ by:

$$\mathbf{E}_\pi(h) \;\approx\; \frac{1}{M-B}\sum_{i=B+1}^{M} h(X_i)\,,$$

  where B ("burn-in") is chosen large enough so $X_B \approx \pi$, and M is chosen large enough to get good Monte Carlo estimates.
- How to design such a Markov chain? One good way is:
- METROPOLIS ALGORITHM (1953):
  - Choose some initial value X_0 (perhaps random).
  - Then, given X_{n-1}, choose a proposal state $Y_n \sim MVN(X_{n-1}, \sigma^2\, I)$ for some fixed $\sigma > 0$ (say).
  - Let $A_n = \pi(Y_n) \,/\, \pi(X_{n-1}) = g(Y_n) \,/\, g(X_{n-1})$, and $U_n \sim \text{Uniform}[0,1]$.
  - Then, if $U_n < A_n$, set $X_n = Y_n$ ("accept"), otherwise set $X_n = X_{n-1}$ ("reject").
  - Repeat, for $n = 1, 2, 3, \ldots, M$.
  - (Note: only need to compute $\pi(Y_n) \,/\, \pi(X_{n-1})$, so the normalising constant c cancels and is not required.)
  - (Why does it work? Markov chain theory – later!)
  - Try it: www.probability.ca/metropolis (Javascript; formerly Java.)
  - Note: This version is called "random walk Metropolis" (RWM). Why? Because the proposals, if we always accepted them, would form a traditional random walk process.
- How large B? Difficult to say! Some theory (later) ... usually just use trial-and-error / statistical analysis of output, and hope for the best ...
- What initial value X_0?
  - Virtually any one will do, but "central" ones best.
  - Can also use an "overdispersed starting distribution": choose X_0 randomly from some distribution that "covers" the "important" parts of the state space. Good for checking consistency ...
- EXAMPLE: $g(y) = y^3 \sin(y^4) \cos(y^5)\, \mathbf{1}_{0<y<1}$.
  - Want to compute (again!) $\mathbf{E}_\pi(h)$ where $h(y) = y^2$.
  - Use Metropolis algorithm with proposal $Y \sim N(X, 1)$. [file "Rmet"]

```
g = function(y) {
    if ( (y<0) || (y>1) ) return(0)
    else return( y^3 * sin(y^4) * cos(y^5) ) }
h = function(y) { return(y^2) }
M = 11000; B = 1000; numaccept = 0;
xlist = rep(NA,M); X = runif(1) # Overdispersed
for (i in 1:M) {
    Y = X + rnorm(1)  # Proposal
    if (runif(1) < g(Y) / g(X) ) { # Accept!
        X = Y; numaccept = numaccept + 1; }
    xlist[i] = X;
}
print( mean(h(xlist[(B+1):M])) )
```

- Works pretty well, but lots of variability!
- Can also investigate the "trace plot":

```
plot(xlist, type='l')
```

- Plot: appears to have "good mixing".
- And the "auto-correlation function":

```
acf(xlist)
```

- acf: has some serial autocorrelation $\mathbf{E}(X_n X_{n+k})$. Important! (Soon.)

- EXAMPLE: $\pi(x_1, x_2) = C\,|\cos(\sqrt{x_1\,x_2})|\; I(0 \le x_1 \le 5,\, 0 \le x_2 \le 4)$.
  - Want to compute $\mathbf{E}_\pi(h)$, where $h(x_1, x_2) = e^{x_1} + (x_2)^2$.
  - Metropolis algorithm (file "Rmet2") ... works, but large uncertainty.

```
g = function(x) {
    if ( (x[1]<0)||(x[1]>5)||(x[2]<0)||(x[2]>4) )
        return(0)
    else return( abs( cos(sqrt(x[1]*x[2])) ) ) }
h = function(x) { return( exp(x[1]) + x[2]^2 ) }
x1list = x2list = rep(NA,M)
X = c(5*runif(1), 4*runif(1)) # overdispersed
for (i in 1:M) {
    Y = X + rnorm(2)  # Proposal
    if (runif(1) < g(Y) / g(X) ) { # Accept!
        X = Y; numaccept = numaccept + 1; }
    x1list[i] = X[1]; x2list[i] = X[2];
}
print( mean(h(xlist[(B+1):M])) )
```

  - Gets between about 34 and 44 ... (Mathematica gets 38.7044)
  - Individual plots appear to have "good mixing" ...
  - Joint plot shows fewer samples where $x_1 x_2 \approx (\pi/2)^2 \doteq 2.5$ ...

- OPTIMAL SCALING:
  - What if we change σ? How does that affect estimate? plot? acf?
  - Can change proposal distribution to $Y_n \sim MVN(X_{n-1},\, \sigma^2 I)$ for any choice of $\sigma > 0$. Which is best?
  - Can experiment: "www.probability.ca/metropolis", "Rmet", "Rmet2".
  - If σ too small, then usually accept, but chain won't move much.

- If σ too large, then will usually reject proposals, so chain still won't move much.
- Optimal: need σ "just right" to avoid both extremes. ("Goldilocks Principle")
- Some theory ... limited ... active area of research ...
- General principle: the acceptance rate should be far from 0 and far from 1.
- Surprising Fact: In a certain idealised high-dimensional limit, optimal acceptance rate is 0.234 (!). [Roberts et al., Ann Appl Prob 1997; Roberts and Rosenthal, Stat Sci 2001] (More later!)

COMPONENTWISE (VARIABLE-AT-A-TIME) MCMC:

- The above algorithm is a "full-dimensional" Metropolis algorithm.
- Alternative: "componentwise" Metropolis algorithm:
  - Propose to move just one coordinate at a time, leaving all the other coordinates fixed. (Easier?)
  - e.g. proposal Y_n has $Y_{n,i} \sim N(X_{n-1,i}, \sigma^2)$, with $Y_{n,j} = X_{n-1,j}$ for $j \neq i$. (Here $Y_{n,i}$ is the i^{th} coordinate of Y_n.)
  - Then accept/reject with usual Metropolis rule ("Componentwise Metropolis", or "Variable-at-a-time Metropolis", or "Metropolis-within-Gibbs").
- Need to choose which coordinate to update each time ...
  - Could choose in sequence $1, 2, \ldots, d, 1, 2, \ldots$ ("systematic-scan").

```
for (i in 1:M) {
  for (coord in 1:d) {
    Y = X
    Y[coord] = X[coord] + sigma * rnorm(1)
    if (runif(1) < g(Y) / g(X))
        X = Y  # accept proposal
    x1list[d*i-d+coord] = X[1]  # etc.
  }
}
```

  - Or, choose $\sim$ Uniform$\{1, 2, \ldots, d\}$ each time ("random-scan").

```
for (i in 1:M) {
  coord = sample(1:d, 1)  # uniform on {1,...,d}
  Y = X
  Y[coord] = X[coord] + sigma * rnorm(1)
  if (runif(1) < g(Y) / g(X))
    X = Y  # accept proposal
  x1list[i] = X[1]  # etc.
  }
}
```

  - Note: one systematic-scan iteration takes as long as d random-scan ones ...
- EXAMPLE: again $\pi(x_1, x_2) = C\,|\cos(\sqrt{x_1\,x_2})|\; I(0 \leq x_1 \leq 5,\, 0 \leq x_2 \leq 4)$, and $h(x_1, x_2) = e^{x_1} + (x_2)^2$. (Recall: Mathematica gives $\mathbf{E}_\pi(h) \doteq 38.7044$.)
  - Works with systematic-scan (file "Rcompwise") or random-scan (file "Rcompwise2"). Better? Worse? Can investigate various plots:

```
plot(x1list, type='l')
plot(x2list, type='l')
plot(x1list, x2list, type='l')
```

MCMC STANDARD ERROR:

- What about MCMC's standard error, i.e. uncertainty?
  - It's usually larger than in the i.i.d. case (due to the positive correlations), and harder to quantify.
- Simplest: re-run the chain many times, with same M and B, with different initial values drawn from some overdispersed starting distribution, and get a fresh estimate each time, and then compute the standard error of the sequence of estimates.
  - Then can analyse the estimates obtained as iid ...
- But how to estimate standard error from a single run?
- i.e., how to estimate $v \equiv \mathbf{Var}\left(\frac{1}{M-B}\sum_{i=B+1}^{M} h(X_i)\right)$?
  - For simplicity, let $\overline{h}(x) = h(x) - \mathbf{E}_\pi(h)$, so $\mathbf{E}_\pi(\overline{h}) = 0$.
  - And, assume B large enough that $X_i \approx \pi$ for $i > B$.
  - Then, for large $M - B$,

$$v \;\approx\; \mathbf{E}_\pi\Big[\Big(\Big[\frac{1}{M-B}\sum_{i=B+1}^{M} h(X_i)\Big] - \mathbf{E}_\pi(h)\Big)^2\Big] \;=\; \mathbf{E}_\pi\Big[\Big(\frac{1}{M-B}\sum_{i=B+1}^{M} \overline{h}(X_i)\Big)^2\Big]$$

$$= \frac{1}{(M-B)^2}\Big[(M-B)\,\mathbf{E}_\pi[\overline{h}(X_i)^2] + 2(M-B-1)\,\mathbf{E}_\pi[\overline{h}(X_i)\overline{h}(X_{i+1})] + 2(M-B-2)\,\mathbf{E}_\pi[\overline{h}(X_i)\overline{h}(X_{i+2})] + \ldots\Big]$$

$$\approx \frac{1}{M-B}\Big(\mathbf{E}_\pi[\overline{h}(X_i)^2] + 2\,\mathbf{E}_\pi[\overline{h}(X_i)\overline{h}(X_{i+1})] + 2\,\mathbf{E}_\pi[\overline{h}(X_i)\overline{h}(X_{i+2})] + \ldots\Big)$$

$$= \frac{1}{M-B}\Big(\mathbf{Var}_\pi(h) + 2\,\mathbf{Cov}_\pi(h(X_i), h(X_{i+1})) + 2\,\mathbf{Cov}_\pi(h(X_i), h(X_{i+2})) + \ldots\Big)$$

$$= \frac{1}{M-B}\,\mathbf{Var}_\pi(h)\Big(1 + 2\,\mathbf{Corr}_\pi(h(X_i), h(X_{i+1})) + 2\,\mathbf{Corr}_\pi(h(X_i), h(X_{i+2})) + \ldots\Big)$$

$$\equiv \frac{1}{M-B}\,\mathbf{Var}_\pi(h)(\text{varfact}) \;=\; (\text{iid variance})\,(\text{varfact})\,,$$

where

$$\text{“varfact”} \;=\; 1 + 2\sum_{k=1}^{\infty} \mathbf{Corr}_\pi\Big(h(X_0), h(X_k)\Big) \;\equiv\; 1 + 2\sum_{k=1}^{\infty} \rho_k$$

with $\rho_k = \mathbf{Corr}_\pi\Big(h(X_0), h(X_k)\Big)$. (Can compute with the R function “acf”.)

- Then standard error $= se = \sqrt{v} = (iid\text{-}se)\,\sqrt{\text{varfact}}$.
- varfact is usually called “integrated auto-correlation time” or “ACT”.
- Since $\rho_0 = 1$, we also have

$$\text{varfact} \;=\; 2\Big(\sum_{k=0}^{\infty} \rho_k\Big) - 1\,.$$

- Aside: Can also assume that the entire chain $\{X_n\}_{n=-\infty}^{\infty}$ is stationary, so $\rho_{-k} = \rho_k$; then we also have

$$\text{varfact} = \sum_{k=-\infty}^{\infty} \rho_k \,.$$

- Can estimate both iid variance, and varfact, from the sample run.

```
se1 =  sd(hlist[(B+1):M]) / sqrt(M-B)
varfact <- function(xxx) { 2 * sum(acf(xxx,
                              plot=FALSE)$acf) - 1 }
thevarfact = varfact(hlist[(B+1):M])
se = se1 * sqrt( thevarfact )
```

- Note: to compute varfact, best not to sum over all k, just "relevant" k.
  - e.g. until, say, $|\rho_k| < 0.05$ or $\rho_k < 0$ or ...
  - R's built-in "acf" function has a "lag.max" parameter.
  - The default is lag.max = $10 \log_{10}(N)$. But better to select lag.max yourself by visual inspectaion. Or write your own version – better!
  - e.g. file "Rmet" and file "Rmet2". (Recall: true answers are about 0.766 and 38.7, respectively.) Also file "Rnorm", file "Rheavy", file "Rheavy2".
  - Usually varfact $\gg 1$; try to get "better" chains so varfact smaller.
  - Sometimes even try to design chain to get varfact < 1 ("antithetic").
  - Work in parallel? (Antithetically??) Some work, but limited. (Project?)

END WEEK #4

CONFIDENCE INTERVALS:

- Suppose we estimate $u \equiv \mathbf{E}_\pi(h)$ by the quantity $e = \frac{1}{M-B} \sum_{i=B+1}^{M} h(X_i)$, and obtain an estimate e and an approximate variance (as above) v.
- Then what is, say, a 95% confidence interval for u?
- Well, if we have a central limit theorem (CLT), then for large $M - B$, we will have $e \approx N(u, v)$.
  - So $(e-u)\, v^{-1/2} \approx N(0,1)$.
  - So, $\mathbf{P}(-1.96 < (e-u)\, v^{-1/2} < 1.96) \approx 0.95$.
  - So, $\mathbf{P}(-1.96\sqrt{v} < e-u < 1.96\sqrt{v}\,) \approx 0.95$.
  - i.e., with probability 95%, the interval $(e - 1.96\sqrt{v},\ e + 1.96\sqrt{v})$ will contain u.
- So, can compute confidence intervals exactly as before, except using the new $se = \sqrt{v} = (iid\text{-}se)\sqrt{\text{varfact}}$ as above.
- e.g. the files Rmet and Rmet2. (Recall: true answers are about 0.766 and 38.7, respectively.)
- But does a CLT even hold?? Usually ...
  - Does not follow from classical i.i.d. CLT. Does not always hold. But often does. See e.g. Chapter 17 of Meyn & Tweedie (1993).
  - For example, CLT holds if chain is "geometrically ergodic" (later!) and $\mathbf{E}_\pi(|h|^{2+\delta}) < \infty$ for some $\delta > 0$.

- (If chain also reversible then don't need δ: Roberts and Rosenthal, "Geometric ergodicity and hybrid Markov chains", ECP 1997.)
- Can get alternative (slightly larger) confidence intervals even without a CLT, if have consistent variance estimator: Rosenthal, "Simple confidence intervals for MCMC without CLTs", EJS 2017; and the recent follow-up paper Jiang et al. (2020).)

SUBSAMPLING (THINNING):

- The autocorrelations (acf) of an MCMC run usually start near 1, and then decrease until they become negligible after some lag L. [file "Rmet"]
  - This means that every L^{th} iteration of the chain is approximately independent, so that $\{X_{B+Ln}\}_{n=1}^{\infty}$ are approximately i.i.d. $\sim \pi$.
  - This could be useful: i.i.d. samples, classical standard error and confidence intervals, good tests for accuracy, etc.
  - The number of samples is reduced from $M-B$ to $(M-B)/L$, which increases the estimator variance by a factor of L. Bad. (Assume for simplicity that $M-B$ is a multiple of L.)
  - But the correlations are reduced (or maybe even zero). Good.
  - So, does it improve the actual estimator, e.g. its variance?
- In fact, subsampling always increases the estimator variance!
  - Assume the burn-in B is large enough to reach stationarity.
  - Then the usual estimator is $e = \frac{1}{M-B}\sum_{i=B+1}^{M} h(X_i)$.
  - Consider the alternative subsampled estimator a given by

$$\frac{1}{(M-B)/L}\left[h(X_{B+1})+h(X_{B+L+1})+h(X_{B+2L+1})+\ldots+h(X_{M-L+1})\right].$$

  - Trick (Maceachern & Berliner, 1994): For $1 \le k \le L$, let

$$y_k \;=\; h(X_{B+k}) + h(X_{B+L+k}) + h(X_{B+2L+k}) + \ldots + h(X_{M-L+k})\,.$$

  - So $a = \frac{1}{(M-B)/L}\, y_1$, and $\mathbf{Var}(a) = \frac{1}{[(M-B)/L]^2}\,\mathbf{Var}(y_1)$.
  - Then, since each $\mathbf{Var}_\pi(y_i) = \mathbf{Var}_\pi(y_1)$ by stationarity,

$$\mathbf{Var}(e) \;\equiv\; v \;=\; \mathbf{Var}\Big(\frac{1}{M-B}\sum_{i=B+1}^{M} h(X_i)\Big)$$

$$= \mathbf{Var}\Big(\frac{1}{M-B}(y_1 + y_2 + \ldots + y_L)\Big)$$

$$= \frac{1}{(M-B)^2}\left[L\;\mathbf{Var}_\pi(y_1) + \sum_{i\neq j}\mathbf{Cov}_\pi(y_i,y_j)\right].$$

  - But $\mathbf{Cov}_\pi(y_i,y_j) = \mathbf{Corr}_\pi(y_i,y_j)\sqrt{\mathbf{Var}_\pi(y_i)\,\mathbf{Var}_\pi(y_j)}$ $\le \sqrt{\mathbf{Var}_\pi(y_i)\,\mathbf{Var}_\pi(y_j)} = \sqrt{\mathbf{Var}_\pi(y_1)\,\mathbf{Var}_\pi(y_1)} = \mathbf{Var}_\pi(y_1)$. So,

$$\mathbf{Var}(e) \;\le\; \frac{1}{(M-B)^2}\left[L\;\mathbf{Var}_\pi(y_1) + L(L-1)\,\mathbf{Var}_\pi(y_1)\right]$$

$$= \frac{1}{(M-B)^2}\;L^2\;\mathbf{Var}_\pi(y_1) \;=\; \frac{1}{[(M-B)/L]^2}\;\mathbf{Var}_\pi(y_1) \;=\; \mathbf{Var}(a)\,.$$

  - This shows that $\mathbf{Var}(a) \ge \mathbf{Var}(e)$.

  - Actually $\mathbf{Var}(a) > \mathbf{Var}(e)$, since will have $\mathbf{Corr}_\pi(y_i, y_j) < 1$.

- Conclusion: Subsampling can only make the estimator variance larger, i.e. subsampling does not improve the estimator accuracy.

- However, if lots of extra computation is required to compute needed functional values $h(X_i)$, then subsampling might help: Owen 2017

- Separate point: Although thinning shouldn't normally be used for estimation, it can still be used when creating trace plots of long runs.
  - Plotting more than 10,000 total values hardly affects a plot's appearance, but it can make the file sizes very large and awkward.
  - For example, if `xlist` has length 1,000,000, then it might be preferable to plot just every 100'th value, with e.g.:

```
plot(xlist[100*(1:10000)], type='l')
```

  - But don't thin below 10,000 total values; that could be deceptive.

JUSTIFICATION OF METROPOLIS ALGORITHM:

- (Uses Markov chain theory ... e.g. STA447/2006 ... already know?)

- Basic fact: If a Markov chain is "irreducible", with "stationarity distribution" π, then it converges as $n \to \infty$. More precisely:

- THEOREM: If a Markov chain is π-irreducible, with stationarity probability density π, then for $B > 0$ and π-a.e. initial value $X_0 = x$, if $\mathbf{E}_\pi(|h|) < \infty$, then $\lim_{M\to\infty} \frac{1}{M-B} \sum_{i=B+1}^{M} h(X_i) = \mathbf{E}_\pi(h)$.

- Let's figure out what this means ...

- Markov chain transition probabilities and notation:
  - Discrete case (e.g. $\mathcal{X} \equiv \{1, \ldots, 6\}$): $P(i,j) = \mathbf{P}(X_{n+1} = j \,|\, X_n = i)$.
  - General/contin case (e.g. $\mathcal{X} = \mathbf{R}$): $P(x, A) = \mathbf{P}(X_{n+1} \in A \,|\, X_n = x)$.
  - Stationary distribution: $\Pi(A) = \sum_{i\in A} \pi(i)$ or $\Pi(A) = \int_A \pi(x)\, dx$.

- Then π-irreducible means that you have positive probability of eventually getting from anywhere to anywhere else with positive π.
  - Discrete case: For all $i, j \in \mathcal{X}$ (the state space) with $\pi(j) > 0$, there is $n \in \mathbf{N}$ such that $\mathbf{P}(X_n = j \,|\, X_0 = i) > 0$.
  - General/continuous case: For all $x \in \mathcal{X}$ and all $A \subseteq \mathcal{X}$ with $\Pi(A) > 0$, there is $n \in \mathbf{N}$ such that $\mathbf{P}(X_n \in A \,|\, X_0 = x) > 0$. (Necessary since often $\mathbf{P}(X_n = y \,|\, X_0 = x) = 0$ for all y.)
  - Irreducibility is usually satisfied for MCMC, i.e. not a problem.

- What about Π being a stationary distribution?
  - This means that if we start with the probabilities Π, and then run the Markov chain for one step, that we will still have the same exact probabilities Π.
  - That is, if $X_0 \sim \pi$, then also $X_1 \sim \pi$.
  - Will this be true for the Metropolis algorithm??

- Begin with the DISCRETE CASE (e.g. www.probability.ca/metropolis).
  - In the discrete case, π being stationary means that if $\mathbf{P}(X_0 = i) = \pi(i)$ for all i, then also $\mathbf{P}(X_1 = j) = \pi(j)$ for all j.
  - But $\mathbf{P}(X_1 = j) = \sum_{i\in S} \mathbf{P}(X_0 = i, X_1 = j) = \sum_{i\in S} \mathbf{P}(X_0 = i)\, P(i,j) = \sum_{i\in S} \pi(i)\, P(i,j)$.

- So, π is stationary if $\sum_{i \in S} \pi(i)\, P(i,j) \;=\; \pi(j)$ for all j.

- One way to check stationarity: reversibility.
  - Definition: A chain is "(time) reversible" if $\pi(i)\, P(i,j) \;=\; \pi(j)\, P(j,i)$ for all $i, j \in \mathcal{X}$. (discrete case)
  - (Intuition: if $X_0 \sim \pi$, i.e. $\mathbf{P}(X_0 = i) = \pi(i)$ for all $i \in \mathcal{X}$, then $\mathbf{P}(X_0 = i,\ X_1 = j) = \pi(i)\, P(i,j) = \mathbf{P}(X_0 = j,\ X_1 = i)\ \ldots\ )$
  - Does reversibility imply that π is a stationary distribution?
  - Yes! Suppose $X_0 \sim \pi$, i.e. that $\mathbf{P}(X_0 = i) = \pi(i)$ for all $i \in \mathcal{X}$.
  - Then by reversibility,

$$\mathbf{P}(X_1 = j) \;=\; \sum_{i \in \mathcal{X}} \mathbf{P}(X_0 = i)\, P(i,j) \;=\; \sum_{i \in \mathcal{X}} \pi(i)\, P(i,j)$$

$$= \sum_{i \in \mathcal{X}} \pi(j)\, P(j,i) \;=\; \pi(j) \sum_{i \in \mathcal{X}} P(j,i) \;=\; \pi(j)\,,$$

    i.e. $X_1 \sim \pi$ too. So, π is a stationary distribution!
  - Will reversibility hold for the Metropolis algorithm?

- Let $q(i,j) = \mathbf{P}(Y_n = j \,|\, X_{n-1} = i)$ be the proposal distribution, e.g. perhaps $q(i, i+1) = q(i, i-1) = 1/2$.
  - Recall that q is symmetric, i.e. $q(i,j) = q(j,i)$ for all $i, j \in \mathcal{X}$.
  - Then if $\alpha(i,j)$ is the probability of accepting a proposed move from i to j, then

$$\alpha(i,j) = \mathbf{P}(U_n < A_n \,|\, X_{n-1} = i,\ Y_n = j)$$

$$= \mathbf{P}(U_n < \frac{\pi(j)}{\pi(i)}) \;=\; \min[1,\ \frac{\pi(j)}{\pi(i)}]\,.$$

    (Assume that $\pi(i) > 0$, otherwise we would never visit i.)

- Then we compute that for $i, j \in \mathcal{X}$ with $i \neq j$,

$$P(i,j) \;=\; \mathbf{P}_i(\text{propose to move to } j,\ \text{then accept it})$$

$$= \; q(i,j)\ \alpha(i,j) \;=\; q(i,j)\ \min(1,\ \frac{\pi(j)}{\pi(i)})\,.$$

  - Hence, we compute that

$$\pi(i)\, P(i,j) \;=\; q(i,j)\ \min(\pi(i),\ \pi(j))\,.$$

  - Then it follows by the symmetry of q that

$$\pi(i)\, P(i,j) \;=\; q(j,i)\ \min(\pi(i),\ \pi(j)) \;=\; \pi(j)\, P(j,i)\,.$$

  - This shows that, if $i \neq j$, then $\pi(i)\, P(i,j) \;=\; \pi(j)\, P(j,i)$.
  - But if $i = j$, then of course $\pi(i)\, P(i,j) \;=\; \pi(j)\, P(j,i)$.

- This shows that any (discrete) Metropolis algorithm is reversible.
  - Hence, any (discrete) Metropolis algorithm has π stationary.
  - So, from the above Theorem, assuming irreduciblity, if $\mathbf{E}_\pi(|h|) < \infty$, then $\lim_{M \to \infty} \frac{1}{M-B} \sum_{i=B+1}^{M} h(X_i) = \mathbf{E}_\pi(h)$.

- Now what about the GENERAL/CONTINUOUS CASE?

- Similar, but we need some more notation:

- Let $\pi(x)$ be the target density function on $\mathcal{X}$ (e.g. on $\mathbf{R}$ or $\mathbf{R}^d$).
- Let $q(x,y)$ be the proposal density function for y from x. (e.g. $q(x,y) = (2\pi\sigma)^{-d/2} \exp\,(-\sum_{i=1}^d (y_i - x_i)^2/2\sigma^2)$.)
- Assume again that q is symmetric: $q(x,y) = q(y,x)$.
- Let $\alpha(x,y) = \min[1,\ \frac{\pi(y)}{\pi(x)}]$ be the probability of accepting a proposed move from x to y.

- Then, similar to before, if $x \notin S$, then

$$P(x,S) \ = \ \mathbf{P}(Y_1 \in S,\ U_1 < A_1 \,|\, X_0 = x)$$

$$= \int_S q(x,y)\ \alpha(x,y)\ dy \ = \ \int_S q(x,y)\ \min[1,\ \pi(y)/\pi(x)]\ dy\,.$$

- Shorthand: write "$P(x,dy)$" for the transition measure, i.e. a quantity whose integral over any subset S is equal to the transition probability to S, meaning that $P(x,S) = \int_{y\in S} P(x,dy)$.
  - Then, for $x \neq y$, $P(x,dy) = q(x,y)\ \min[1,\ \pi(y)/\pi(x)]\,dy$.
  - Hence, for $x \neq y$,

$$\pi(x)\,dx\,P(x,dy) \ = \ \pi(x)\,dx\ q(x,y)\ \min[1,\ \pi(y)/\pi(x)]\,dy$$

$$= \ q(x,y)\ \min[\pi(x),\ \pi(y)]\,dy\,dx\,.$$

  - This is symmetric in x and y, which shows that $\pi(x)\,dx\,P(x,dy) = \pi(y)\,dy\,P(y,dx)$ for all $x \neq y$.
  - But that is of course true if $x = y$.
  - So, $\pi(x)\,dx\,P(x,dy) = \pi(y)\,dy\,P(y,dx)$ for all $x, y \in \mathcal{X}$. ("reversible")

- How does "reversible" help? Just like in the discrete case!

- Indeed, suppose $X_0 \sim \pi$, i.e. we "start in stationarity". Then

$$\mathbf{P}(X_1 \in S) \ = \ \int_{x\in\mathcal{X}} \int_{y\in S} \pi(x)\,dx\,P(x,dy) \ = \ \int_{x\in\mathcal{X}} \int_{y\in S} \pi(y)\,dy\,P(y,dx)$$

$$= \ \int_{y\in S} \pi(y)\,dy \int_{x\in\mathcal{X}} P(y,dx) \ = \ \int_{y\in S} \pi(y)\,dy\,,$$

  so also $X_1 \sim \pi$. So, chain "preserves" π, i.e. π is stationary.

- And, again, almost always irreducible, so the Theorem applies.

- So, again, if $\mathbf{E}_\pi(|h|) < \infty$, then $\lim_{M\to\infty} \frac{1}{M-B} \sum_{i=B+1}^M h(X_i) = \mathbf{E}_\pi(h)$.

EXAMPLES RE JUSTIFICATION OF METROPOLIS:

- EXAMPLE #1: Metropolis algorithm where $\mathcal{X} = \mathbf{Z}$, $\pi(x) = 2^{-|x|}/3$, and $q(x,y) = \frac{1}{2}$ if $|x-y| = 1$, otherwise 0.
  - Reversible? Yes, it's a Metropolis algorithm!
  - π stationary? Yes, follows from reversibility!
  - Irreducible? Yes: $\pi(x) > 0$ for all $x \in \mathcal{X}$, so can get from x to y in $|x-y|$ steps.
  - So, by theorem, probabilities and expectations converge to those of π – good.

- EXAMPLE #2: Same as #1, except now $\pi(x) = 2^{-|x|-1}$ for $x \neq 0$, with $\pi(0) = 0$.

- Still reversible, π stationary, same as before.
- Irreducible? No – can't go from positive to negative!

- EXAMPLE #3: Same as #2, except now $q(x, y) = \frac{1}{4}$ if $1 \leq |x - y| \leq 2$, otherwise 0.
  - Still reversible, π stationary, same as before.
  - Irreducible? Yes – can "jump over 0" to get from positive to negative, and back!
- EXAMPLE #4: Metropolis algorithm with $\mathcal{X} = \mathbf{R}$, and $\pi(x) = C\, e^{-x^6}$, and proposals $Y_n \sim \text{Uniform}[X_{n-1} - 1, X_{n-1} + 1]$.
  - Reversible? Yes since it's Metropolis, and $q(x, y)$ still symmetric.
  - π stationary? Yes since reversible!
  - Irreducible? Yes, since the n-step transitions $P^n(x, dy)$ have positive density whenever $|y - x| < n$.
  - So, by theorem, probabilities and expectations converge to those of π – good.
- EXAMPLE #5: Same as #4, except now $\pi(x) = C_1\, e^{-x^6}(\mathbf{1}_{x<2} + \mathbf{1}_{x>4})$.
  - Still reversible and stationary, same as before.
  - But no longer irreducible: cannot jump from $[4, \infty)$ to $(-\infty, 2]$ or back.
  - So, does not converge.
- EXAMPLE #6: Same as #5, except now proposals are $Y_n \sim \text{Uniform}[X_{n-1} - 5, X_{n-1} + 5]$.
  - Still reversible and stationary, same as before.
  - And now irreducible, too: now can jump from $[4, \infty)$ to $(-\infty, 2]$ or back.
- EXAMPLE #7: Same as #6, except now $Y_n \sim \text{Uniform}[X_{n-1} - 5, X_{n-1} + 10]$.
  - Makes no sense – proposals not symmetric, so not a Metropolis algorithm! (Not even symmetrically zero, for the Metropolis-Hastings algorithm below, e.g. have positive density $3 \to 9$ but not $9 \to 3$.)

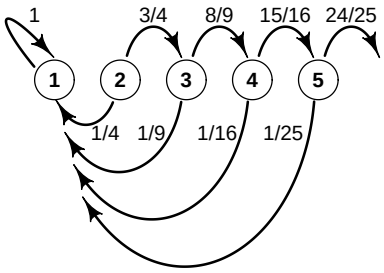
JUSTIFICATION FOR COMPONENTWISE METROPOLIS:

- The exact same justification works just like for the "regular" (full-dimensional) Metropolis algorithm:
  - If we update the variables one-at-a-time (componentwise), then each individual step is still reversible (for the same reason).
  - So, π is stationary for each individual step.
  - So, π is stationary for the combined steps, too. (Though not necessarily reversible for the combined steps.)
  - And, if the steps are combined appropriately (e.g. systematic-scan or random-scan), so that each component eventually moves, then it will still be π-irreducible, too.
  - So then, like any π-irreducible Markov chain with stationary distribution π, it will eventually converge to π, so it is still valid.

END WEEK #5

INITIAL DISTRIBUTION CONDITION:

- Why does the above Theorem say "π-a.e." $X_0 = x$?
  - It means that convergence holds from "almost every" initial state, but there could still be certain "bad" initial states (having total stationary measure 0) from which it fails.
- Example: $\mathcal{X} = \{1, 2, 3, 4, \ldots\}$, and $P(1, \{1\}) = 1$, and for $x \geq 2$, $P(x, \{1\}) = 1/x^2$ and $P(x, \{x+1\}) = 1 - (1/x^2)$.



  - Stationary distribution?
    $\Pi(\cdot) = \delta_1(\cdot)$, i.e. $\Pi(S) = \mathbf{1}_{1 \in S}$ for $S \subseteq \mathcal{X}$.
  - π-irreducible?
    Yes, since if $\Pi(S) > 0$ then $1 \in S$ so $P(x, S) \geq P(x, \{1\}) > 0$ for all $x \in \mathcal{X}$.
  - Converges?
    Yes, by Theorem, for π-a.e. X_0, have $\lim_{n\to\infty} \mathbf{P}(X_n \in S) = \Pi(S)$, i.e. $\lim_{n\to\infty} \mathbf{P}(X_n = 1) = 1$.
  - From everywhere?
    No! If $X_0 = x \geq 2$, then $\mathbf{P}[X_n = x + n \text{ for all } n] = \prod_{j=x}^{\infty}[1 - (1/j^2)]$. This is actually > 0, by "infinite product theory" since $\sum_{j=x}^{\infty}(1/j^2) < \infty$. So, $\lim_{n\to\infty} \mathbf{P}(X_n = 1) \neq 1$.
  - Convergence holds if $X_0 = 1$ (which is π-a.e. since $\Pi\{1\} = 1$), but not from $X_0 = x \geq 2$ (which is okay since $\Pi\{2, 3, 4, \ldots\} = 0$).
- So, convergence subtle. But usually holds from any $x \in \mathcal{X}$. ("Harris recurrent", see e.g. http://probability.ca/jeff/ftpdir/harris.pdf)

METROPOLIS-HASTINGS ALGORITHM:

- The Metropolis algorithm doesn't always work well.
  - Sometimes other MCMC algorithms can help too.
  - With above theory, can derive other valid algorithms!
- Metropolis algorithm works provided the proposal distribution is symmetric, i.e. $q(x, y) = q(y, x)$ for all x, y.
  - So, could replace $Y_n \sim N(X_{n-1}, \sigma^2)$ by e.g. $Y_n \sim \text{Uniform}[X_{n-1} - 1, X_{n-1} + 1]$, or (on discrete space) $Y_n = X_{n-1} \pm 1$ prob. $\frac{1}{2}$ each, etc.
  - But what if q is not symmetric? Can we "fix" the algorithm?
- Hastings, Biometrika 1970 [Canadian! see www.probability.ca/hastings]:
  - Claim: If we replace "$A_n = \pi(Y_n) \,/\, \pi(X_{n-1})$" by $A_n = \frac{\pi(Y_n)\, q(Y_n, X_{n-1})}{\pi(X_{n-1})\, q(X_{n-1}, Y_n)}$, then the algorithm is still valid even if q is not symmetric.
  - That is, we still accept if $U_n < A_n$, otherwise reject.
  - (Intuition: if $q(x, y) >> q(y, x)$, then Metropolis chain would spend too much time at y and not enough at x, so need to accept fewer moves $x \to y$.)

- Though we do require that $q(x, y) > 0$ iff $q(y, x) > 0$.
- Then Metropolis is special case where $\frac{\pi(Y_n)\, q(Y_n, X_{n-1})}{\pi(X_{n-1})\, q(X_{n-1}, Y_n)} = \frac{\pi(Y_n)}{\pi(X_{n-1})}$.

- Can we modify the above proof to work for Metropolis-Hastings, too?
- For Metropolis, key was that the Markov chain is reversible, i.e. $\pi(x)\, P(x, y) = \pi(y)\, P(y, x)$, i.e. $q(x, y)\, \alpha(x, y)\, \pi(x)$ is symmetric in x and y.
  - If instead $A_n = \frac{\pi(Y_n)\, q(Y_n, X_{n-1})}{\pi(X_{n-1})\, q(X_{n-1}, Y_n)}$, i.e. acceptance prob. $\equiv \alpha(x, y) = \min\left[1, \frac{\pi(y)\, q(y,x)}{\pi(x)\, q(x,y)}\right]$, then:

$$q(x, y)\, \alpha(x, y)\, \pi(x) = q(x, y)\ \min\left[1,\ \frac{\pi(y)\, q(y, x)}{\pi(x)\, q(x, y)}\right] \pi(x)$$

$$= \min\left[\pi(x)\, q(x, y),\ \pi(y)\, q(y, x)\right].$$

  - So, $\pi(x)\, P(x, y)$ is still symmetric, even if q wasn't.
  - So, still reversible. So, still have stationary distribution Π.
  - So, if irreducible (nearly always true), then can again apply the Theorem, and again conclude that it converges to π.
- EXAMPLE: again $\pi(x_1, x_2) = C\, |\cos(\sqrt{x_1\, x_2})|\ I(0 \le x_1 \le 5,\ 0 \le x_2 \le 4)$, and $h(x_1, x_2) = e^{x_1} + (x_2)^2$. (Mathematica gives $\mathbf{E}_\pi(h) \doteq 38.7044$.)
  - Proposal distribution: $Y_n \sim MVN(X_{n-1},\ \sigma^2\, (1 + |X_{n-1}|^2)^2\, I)$.
  - (Intuition: larger proposal variance if farther from center.)
  - So, $q(x, y) = C\, (1 + |x|^2)^{-2}\, \exp(-|y - x|^2\, /\, 2\, \sigma^2 (1 + |x|^2)^2)$.
  - Then, can run Metropolis-Hastings algorithm. [file "RMH"]
  - Usually get between 34 and 43, with claimed standard error ≈ 2.
- Can also do Metropolis-Hastings one component at a time, just like for Metropolis. ("Componentwise Metropolis-Hastings", or "Variable-at-a-time Metropolis-Hastings", or "Metropolis-Hastings-within-Gibbs".)

INDEPENDENCE SAMPLER:

- Special case of the Metropolis-Hastings algorithm, so it still converges.
- Propose $\{Y_n\} \sim q(\cdot)$, i.e. the $\{Y_n\}$ are i.i.d. from some fixed density q, independent of X_{n-1}. (e.g. $Y_n \sim MVN(0,\ I_d)$)
  - That is, $q(x, y)$ becomes just $q(y)$, the same for all x.
  - So, $A_n = \frac{\pi(Y_n)\, q(Y_n, X_{n-1})}{\pi(X_{n-1})\, q(X_{n-1}, Y_n)}$ becomes $A_n = \frac{\pi(Y_n)\, q(X_{n-1})}{\pi(X_{n-1})\, q(Y_n)}$.
  - So, accept if $U_n < A_n$ where $U_n \sim \text{Uniform}[0, 1]$ and $A_n = \frac{\pi(Y_n)\, q(X_{n-1})}{\pi(X_{n-1})\, q(Y_n)}$.
- One very special case is: if $q(y) \equiv \pi(y)$, i.e. propose exactly from target density π, then $A_n \equiv 1$.
  - That is, if you make great proposals, then you will always accept them (iid). Makes sense!
- e.g. independence sampler with $\pi(x) = e^{-x}$ and $q(y) = k e^{-ky}$ for $x > 0$.
  - Then if $X_{n-1} = x$ and $Y_n = y$, then $A_n = \frac{e^{-y}\, k e^{-kx}}{e^{-x}\, k e^{-ky}} = e^{(k-1)(y-x)}$. [file "Rind"]
  - $k = 1$: iid sampling (great).
  - $k = 0.01$: proposals way too large (so-so).
  - $k = 5$: proposals somewhat too small (terrible).

- And with $k = 5$, confidence intervals often miss 1. [file "Rind2"]
- Why is large k so much worse than small k? (Later!)

LANGEVIN ALGORITHM:

- Special case of Metropolis-Hastings algorithm.
- $Y_n \sim MVN(X_{n-1} + \frac{1}{2}\,\sigma^2\,\nabla \log \pi(X_{n-1}),\ \sigma^2 I)$.
- Intuition: tries to move in direction where π increasing.
- Based on discrete approximation to "Langevin diffusion".
- Usually more efficient, but requires knowledge and computation of $\nabla \log \pi$, which could be difficult.
- Fortunately, if $\pi(x) = c\,g(x)$, then $\nabla \log \pi(x) = \nabla[\log(c) + \log(g(x))] = \nabla \log(g(x))$, so we still don't need the normalising constant c.
- For theory, see e.g. Roberts & Tweedie, Bernoulli **2(4)**, 341–363, 1996; Roberts & Rosenthal, JRSSB **60**, 255–268, 1998.

- So, lots of MCMC algorithms to choose from.
  - Why do we need them all?
  - To compute with complicated models! For example ...

BAYESIAN STATISTICS:

- Have unknown parameter(s) θ, and a statistical model (likelihood function) for how the distribution of the data Y depends on θ: $\mathcal{L}(Y \,|\, \theta)$.
- Have a <u>prior</u> distribution, representing our "initial" (subjective?) probabilities for θ: $\mathcal{L}(\theta)$.
- Combining these gives a full joint distribution for θ and Y, i.e. $\mathcal{L}(\theta, Y)$.
- Then <u>posterior</u> distribution of θ, $\pi(\theta)$, is then the <u>conditional</u> distribution of θ, <u>conditioned</u> on the observed data y, i.e. $\pi(\theta) = \mathcal{L}(\theta \,|\, Y = y)$.
  - In terms of densities, if have prior density $f_\theta(\theta)$, and likelihood $f_{Y|\theta}(y, \theta)$, then joint density is $f_{\theta,Y}(\theta, y) = f_\theta(\theta)\ f_{Y|\theta}(y, \theta)$, and posterior density is

$$\pi(\theta) \;=\; \frac{f_{\theta,Y}(\theta, y)}{f_Y(y)} \;=\; C\, f_{\theta,Y}(\theta, y) \;=\; C\, f_\theta(\theta)\ f_{Y|\theta}(y, \theta)\,.$$

- Bayesian Statistics Example: VARIANCE COMPONENTS MODEL (a.k.a. "random effects model"):

$$\begin{array}{ccccc} & & \mu & & \\ & \swarrow & \downarrow & \searrow & \\ & \theta_1 & \ldots \quad \ldots & \theta_K & \qquad \theta_i \sim N(\mu, V) \\ \swarrow & \downarrow & & \downarrow & \searrow \\ Y_{11}, \ldots, Y_{1J_1} & \ldots & \ldots & Y_{K1}, \ldots, Y_{KJ_K} & \qquad Y_{ij} \sim N(\theta_i, W) \quad \text{[observed]} \end{array}$$

  - Suppose some population has overall mean μ (unknown).
  - Population consists of K groups.
  - Observe $Y_{i1}, \ldots, Y_{iJ_i}$ from group i, for $1 \le i \le K$.
  - Assume $Y_{ij} \sim N(\theta_i, W)$ (cond. ind.), where θ_i and W unknown.
  - Assume the different θ_i are "linked" by $\theta_i \sim N(\mu, V)$ (cond. ind.), with μ and V also unknown.

- Want to estimate some or all of $V, W, \mu, \theta_1, \ldots, \theta_K$.
- Bayesian approach: use prior distributions, e.g. ("conjugate"):

$$V \sim IG(a_1, b_1); \qquad W \sim IG(a_2, b_2); \qquad \mu \sim N(a_3, b_3)$$

(indep), where a_i, b_i are known constants, and $IG(a, b)$ is the "inverse gamma" distribution, with density $\frac{b^a}{\Gamma(a)}\, e^{-b/x}\, x^{-a-1}$ for $x > 0$. [Here a is the "shape" parameter, and b is the "scale" parameter. Note that if `X=rgamma(shape=a,rate=b)` in R, then $1/X \sim IG(a, b)$.]

- Combining the above dependencies, we see that the joint density is (for $V, W > 0$):

$$\begin{aligned} & f(V, W, \mu, \theta_1, \ldots, \theta_K, Y_{11}, Y_{12}, \ldots, Y_{KJ_K}) \\ = & \left(\frac{b_1^{a_1}}{\Gamma(a_1)} e^{-b_1/V} V^{-a_1-1}\right)\left(\frac{b_2^{a_2}}{\Gamma(a_2)} e^{-b_2/W} W^{-a_2-1}\right)\left(\frac{1}{\sqrt{2\pi b_3}} e^{-(\mu-a_3)^2/2b_3}\right) \times \\ & \times \left(\prod_{i=1}^{K} \frac{1}{\sqrt{2\pi V}} e^{-(\theta_i-\mu)^2/2V}\right)\left(\prod_{i=1}^{K}\prod_{j=1}^{J_i} \frac{1}{\sqrt{2\pi W}} e^{-(Y_{ij}-\theta_i)^2/2W}\right) \\ = & C_2\, e^{-b_1/V} V^{-a_1-1} e^{-b_2/W} W^{-a_2-1} e^{-(\mu-a_3)^2/2b_3} V^{-K/2} W^{-\frac{1}{2}\sum_{i=1}^K J_i} \times \\ & \times \exp\left[-\sum_{i=1}^{K}(\theta_i-\mu)^2/2V\right] \exp\left[-\sum_{i=1}^{K}\sum_{j=1}^{J_i}(Y_{ij}-\theta_i)^2/2W\right]. \end{aligned}$$

  - (Note that the data $\{Y_{ij}\}$, and the prior parameters $\{a_i, b_i\}$, are all treated as constants.)

- Then

$$\begin{aligned} & \pi(V, W, \mu, \theta_1, \ldots, \theta_K) \\ = & f(V, W, \mu, \theta_1, \ldots, \theta_K, Y_{11}, Y_{12}, \ldots, Y_{KJ_K}) \;/\; f_Y(Y_{11}, Y_{12}, \ldots, Y_{KJ_K}) \\ \propto & f(V, W, \mu, \theta_1, \ldots, \theta_K, Y_{11}, Y_{12}, \ldots, Y_{KJ_K}) \\ = & C_3\, e^{-b_1/V} V^{-a_1-1} e^{-b_2/W} W^{-a_2-1} e^{-(\mu-a_3)^2/2b_3} V^{-K/2} W^{-\frac{1}{2}\sum_{i=1}^K J_i} \times \\ & \times \exp\left[-\sum_{i=1}^{K}(\theta_i-\mu)^2/2V\right] \exp\left[-\sum_{i=1}^{K}\sum_{j=1}^{J_i}(Y_{ij}-\theta_i)^2/2W\right]. \end{aligned}$$

- NOTE: Many applications of variance components model, e.g.:
  - Predicting success at law school (D. Rubin, JASA 1980), $K = 82$ schools.
  - Melanoma (skin cancer) recurrence (http://www.mssanz.org.au/MODSIM07/papers/52_s24/Analysing_Clinicals24_Bartolucci_.pdf), with $K = 19$ different patient categories.
  - Comparing baseball home-run hitters (J. Albert, The American Statistician 1992), $K = 12$ players.
  - Analysing fabric dyes (Davies 1947; Box/Tiao 1973; Gelfand/Smith JASA 1990), $K = 6$ batches of dyestuff, $J_i \equiv 5$. (data in file "Rdye")

END WEEK #6

- Here, the dimension is $d = K + 3$, e.g. $K = 19$, $d = 22$. High!
- How to compute/estimate, say, $\mathbf{E}_\pi(W/V)$, or the effect of changing b_1?

- Numerical integration? No, too high-dimensional!
- Importance sampling? Perhaps, but what "f"? Too inefficient!
- Rejection sampling? What "f"? What "K"? Virtually no samples!
- Perhaps MCMC can work!
- But need clever, useful MCMC algorithms!
- Perhaps Metropolis, or ...

- ASIDE: For big complicated π, often better to work with logarithms, e.g. accept iff $\log(U_n) < \log(A_n) = \log(\pi(Y_n)) - \log(\pi(X_{n-1}))$.
  - Then only need to compute $\log(\pi(x))$; helps avoid overflow problems.
  - So, better to program on log scale: $\log \pi(V, W, \mu, \theta_1, \ldots, \theta_K) = \ldots$.
  - Also sometimes simpler, e.g. if $\pi(x) = \exp\left(\sum_{i<j} |x_j - x_i|\right)$, then $\log(\pi(x)) = \sum_{i<j} |x_j - x_i|$. (Best to type in the log formula directly.)

GIBBS SAMPLER:

- (Special case of Componentwise Metropolis-Hastings.)
- Proposal distribution for i^{th} coordinate is equal to the full conditional distribution of that coordinate (according to π), conditional on the current values of all the other coordinates.
  - Can use either systematic or random scan, just like above.
  - Then, always accept. Why? Later!
  - (Intuition: if start in stationary distribution, then update one coordinate from its conditional stationary distribution (and always accept), then the distribution remains the same, i.e. stationarity is preserved.)
- EXAMPLE: Variance Components Model:
  - Update of μ (say) should be from conditional density of μ, conditional on current values of all the other coordinates: $\mathcal{L}(\mu \,|\, V, W, \theta_1, \ldots, \theta_K, Y_{11}, \ldots, Y_{J_K K})$.
  - This conditional density is proportional to the full joint density, but with all variables besides μ treated as constant.
  - Recall: full joint density is:

$$= C_3 e^{-b_1/V} V^{-a_1-1} e^{-b_2/W} W^{-a_2-1} e^{-(\mu-a_3)^2/2b_3} V^{-K/2} W^{-\frac{1}{2}\sum_{i=1}^K J_i} \times$$

$$\times \exp\left[-\sum_{i=1}^{K} (\theta_i - \mu)^2/2V\right] \exp\left[-\sum_{i=1}^{K}\sum_{j=1}^{J_i} (Y_{ij} - \theta_i)^2/2W\right] .$$

  - So, combining "constants" (w.r.t. μ), the conditional density of μ is

$$C_4\, e^{-(\mu-a_3)^2/2b_3} \exp\left[-\sum_{i=1}^{K} (\theta_i - \mu)^2/2V\right] .$$

  - This equals (check!)

$$C_5 \exp\Big(-\mu^2 (\frac{1}{2b_3} + \frac{K}{2V}) + \mu(\frac{a_3}{b_3} + \frac{1}{V}\sum_{i=1}^{K} \theta_i)\Big) .$$

  - Side calculation: if $\mu \sim N(m, v)$, then density $\propto e^{-(\mu-m)^2/2v} \propto e^{-\mu^2(1/2v) + \mu(m/v)}$.

- Hence, here $\mu \sim N(m, v)$, where $1/2v = \frac{1}{2b_3} + \frac{K}{2V}$ and $m/v = \frac{a_3}{b_3} + \frac{1}{V}\sum_{i=1}^{K} \theta_i$.
- Solve: $v = b_3 V/(V + Kb_3)$, and $m = (a_3 V + b_3 \sum_{i=1}^{K} \theta_i) / (V + Kb_3)$.
- So, in Gibbs Sampler, each time μ is updated, we sample it from $N(m, v)$ for this m and v (and always accept).

- Similarly (check!), conditional distribution for V is:

$$C_6 e^{-b_1/V} V^{-a_1-1} V^{-K/2} \exp\left[-\sum_{i=1}^{K} (\theta_i - \mu)^2/2V\right], \qquad V > 0\,.$$

- Recall that "$IG(r, s)$" has density $\frac{s^r}{\Gamma(r)}\, e^{-s/x}\, x^{-r-1}$ for $x > 0$.
- So, conditional distribution for V equals $IG(a_1 + K/2,\ b_1 + \frac{1}{2}\sum_{i=1}^{K}(\theta_i - \mu)^2)$.

- Can similarly compute conditional distributions for W and θ_i. [HW!]
- The systematic-scan Gibbs sampler then proceeds by:
  - Update V from its conditional distribution $IG(\ldots, \ldots)$.
  - Update W from its conditional distribution $IG(\ldots, \ldots)$.
  - Update μ from its conditional distribution $N(\ldots, \ldots)$.
  - Update θ_i from its conditional distribution $N(\ldots, \ldots)$, for $i = 1, 2, \ldots, K$.
  - Repeat all of the above M times.
- Or, the random-scan Gibbs sampler proceeds by choosing one of $V, W, \mu, \theta_1, \ldots, \theta_K$ uniformly at random, and then updating that coordinate from its corresponding conditional distribution.
  - Then repeat this step M times [or $M(K + 3)$ times?].
- How well does it work? Good question! [HW!]

JUSTIFICATION OF GIBBS SAMPLER:

- Special case of Componentwise Metropolis-Hastings:
  - Proposal distribution for i^{th} coordinate is equal to the conditional distribution of that coordinate (according to π), conditional on the current values of all the other coordinates.
  - So, $q_i(x, y)$ is proportional to $\pi(y)\, \mathbf{1}_{y^{(-i)} = x^{(-i)}}$, where $x^{(-i)}$ means all coordinates of x except the i^{th} one.
  - And, the constant of proportionality only depends on $x^{(-i)}$.
  - That is, $q_i(x, y) = C[x^{(-i)}]\, \pi(y)\, \mathbf{1}_{y^{(-i)} = x^{(-i)}}$.
  - So, if $q_i(x, y) > 0$, then $x^{(-i)} = y^{(-i)}$, hence $C[x^{(-i)}] = C[y^{(-i)}]$.
  - But then $A_n = \frac{\pi(Y_n)\, q_i(Y_n, X_{n-1})}{\pi(X_{n-1})\, q_i(X_{n-1}, Y_n)} = \frac{\pi(Y_n)\, C[Y_n^{(-i)}]\, \pi(X_{n-1})}{\pi(X_{n-1})\, C[X_{n-1}^{(-i)}]\, \pi(Y_n)} = 1$.
  - So, it always accepts (i.e., we can ignore the accept-reject step).
- Intuition: If we start with $X_0 \sim \pi$, and then update one component of X from its correct conditional distribution, then we will still have $X_1 \sim \pi$, so π is stationary for the Gibbs sampler.

MCMC CONVERGENCE RATES THEORY:

- $\{X_n\}$: Markov chain on $\mathcal{X}$, with stationary distribution $\Pi(\cdot)$.
- Let $P^n(x, S) = \mathbf{P}[X_n \in S \,|\, X_0 = x]$ be the probabilties for the Markov chain after n steps, when started at x.
  - Hope that for large n, $P^n(x, S) \approx \Pi(S)$.
- Let $D(x, n) = \|P^n(x, \cdot) - \Pi(\cdot)\| \equiv \sup_{S \subseteq \mathcal{X}} |P^n(x, S) - \Pi(S)|$.
  - Hope that $D(x, n) \to 0$. True if π stationary and the chain is π-irreducible and "aperiodic" (almost always holds). But how quickly?
- DEFN: A quantitative bound on convergence is an actual number n^* such that $D(x, n^*) < 0.01$ (say). [If so, then we sometimes say that "the chain converges within n^* iterations".]
  - Quantitative bounds usually very difficult (though I've worked on them a lot, see e.g. Rosenthal, "Quantitative convergence rates of Markov chains: A simple account", Elec Comm Prob 2002 and the references therein). But easier is:
- DEFN: A chain is geometrically ergodic if there is $\rho < 1$ and $M(x) < \infty$ such that $D(x, n) \leq M(x)\, \rho^n$ for π-a.e. $x \in \mathcal{X}$ and all $n \in \mathbf{N}$.
- Fact (mentioned earlier): Central Limit Theorem holds for $\frac{1}{n} \sum_{i=1}^{n} h(X_i)$ if chain is geometrically ergodic and $\mathbf{E}_\pi(|h|^{2+\delta}) < \infty$ for some $\delta > 0$.
  - (If chain also reversible then don't need δ: Roberts and Rosenthal, "Geometric ergodicity and hybrid Markov chains", ECP 1997.)
  - ("Periodic version" of this CLT fact: Roberts and Rosenthal, Probability Surveys 2004, Proposition 30.)
  - If CLT holds, then (as before) have 95% confidence interval $(e - 1.96\sqrt{v},\ e + 1.96\sqrt{v})$, where $v \approx \frac{1}{M-B} \mathbf{Var}_\pi(h)$(varfact).
- One quantitative Theorem: If there is $\delta > 0$ such that $p(x, y) \geq \delta\, \pi(y)$ [prob. or dens.] for all $x, y \in \mathcal{X}$, then $D(x, n) \leq (1 - \delta)^n$ for all $x \in \mathcal{X}$.
  - ("minorisation condition", "uniform ergodicity"; proof by "coupling": Roberts and Rosenthal, Probability Surveys 2004, Theorem 8)
  - Shows geometric ergodicity, together with a quantitative bound.
- Special Case: INDEPENDENCE SAMPLER (mentioned earlier):
  - Proposals $\{Y_n\}$ i.i.d. from some fixed density $q(y)$.
  - Special case of Metropolis-Hastings, where $q(x, y) = q(y)$ depends only on y. So, Π is a stationary distribution.
  - Also, it is π-irreducible provided that $q(x) > 0$ whenever $\pi(x) > 0$.
  - So, by our main Theorem, it will converge to π.
  - But does that guarantee that it will work well?
  - No, e.g. previous "Rind" example with $k = 5$: it still converges (of course), but it performs terribly.
- Suppose an independence sampler satisfies the following condition: There is $\delta > 0$ such that $q(x) \geq \delta\, \pi(x)$ for all $x \in \mathcal{X}$.
  - Then we compute that

$$p(x, y) \;\geq\; q(y)\, \min[1, \frac{\pi(y)\, q(x)}{\pi(x)\, q(y)}] \;=\; \min[q(y),\, \pi(y)\, (q(x) \,/\, \pi(x))]$$

$$\geq\; \min[\delta\, \pi(y),\ \delta\, \pi(y)] \;=\; \delta\, \pi(y)\,.$$

- Hence, from the above Theorem, $D(x,n) \le (1-\delta)^n$ for all $x \in \mathcal{X}$.
- Good, quantitative bound on convergence!

• PREVIOUS EXAMPLE: Independence sampler with $\pi(x) = e^{-x}$ and $q(x) = ke^{-kx}$ for $x > 0$. [file "Rind"]

- If $0 < k \le 1$, then setting $\delta = k$, we have that $q(x) = ke^{-kx} \ge ke^{-x} = k\pi(x) = \delta\pi(x)$ for all $x > 0$, so it's geometrically ergodic, and furthermore $D(x,n) \le (1-k)^n$.
- e.g. if $k = 0.01$, then $D(x, 459) \le (0.99)^{459} \doteq 0.0099 < 0.01$ for all $x > 0$, i.e. the chain "converges within 459 iterations".
- But if $k > 1$, then cannot find any $\delta > 0$ such that $q(x) \ge \delta\pi(x)$ for all x. In fact it is not geometrically ergodic, and for $k > 2$ there is no CLT (Roberts, J. Appl. Prob. **36**, 1210–1217, 1999).
- So, if $k = 5$, then it is not geometrically ergodic, and CLT does not hold. Indeed, confidence intervals often miss 1. [file "Rind2"]
- Fact: if $k = 5$, then $D(0,n) > 0.01$ for all $n \le 4,000,000$, while $D(0,n) < 0.01$ for all $n \ge 14,000,000$, i.e. "converges in between 4 million and 14 million iterations". Slow! [Roberts and Rosenthal, "Quantitative Non-Geometric Convergence Bounds for Independence Samplers", MCAP 2011.]

• What about other MCMC algorithms (besides independence sampler)?

• FACT: If state space is finite, and chain is irreducible and "aperiodic" (which almost always holds), then always geometrically ergodic. (See e.g. J.S. Rosenthal, SIAM Review 37:387-405, 1995.)

• What about for the "random-walk Metropolis algorithm" (RWM), i.e. where $\{Y_n - X_{n-1}\} \sim q$ (i.i.d.) for some fixed symmetric density q?

- e.g. $Y_n \sim N(X_{n-1}, \sigma^2 I)$, or $Y_n \sim \text{Uniform}[X_{n-1} - \delta,\, X_{n-1} + \delta]$.

• FACT: RWM is geometrically ergodic essentially if and only if π has exponentially light tails, i.e. there are $a, b, c > 0$ such that $\pi(x) \le ae^{-b|x|}$ whenever $|x| > c$. (Requires a few technical conditions: π and q continuous and positive; q has finite first moment; and π non-increasing in the tails, with (in higher dims) bounded Gaussian curvature.) [Mengersen and Tweedie, Ann Stat 1996; Roberts and Tweedie, Biometrika 1996]

END WEEK #7

• EXAMPLES: RWM on $\mathbf{R}$ with usual proposals: $Y_n \sim N(X_{n-1}, \sigma^2)$:

→ EXAMPLE #1: $\Pi = N(5, 4^2)$, and functional $h(y) = y^2$, so $\mathbf{E}_\pi(h) = 5^2 + 4^2 = 41$. [file "Rnorm" ... $\sigma = 1$ v. $\sigma = 4$ v. $\sigma = 16$]

- Geometrically ergodic?
  Yes! (By above.)
- Does CLT hold?
  Yes! (geometrically ergodic, and $\mathbf{E}_\pi(|h|^p) < \infty$ for all p.)
- Indeed, confidence intervals "usually" contain 41. [file "Rnorm2"]

→ EXAMPLE #2: $\pi(y) = c\,\frac{1}{(1+y^4)}$, and functional $h(y) = y^2$, so

$$\mathbf{E}_\pi(h) \;=\; \frac{\int_{-\infty}^{\infty} y^2\, \frac{1}{(1+y^4)}\, dy}{\int_{-\infty}^{\infty} \frac{1}{(1+y^4)}\, dy} \;=\; \frac{\pi/\sqrt{2}}{\pi/\sqrt{2}} \;=\; 1\,.$$

- Not exponentially light tails, so not geometrically ergodic; estimates less stable, confidence intervals often miss 1. [file "Rheavy"]

→ EXAMPLE #3: $\pi(y) = \frac{1}{\pi(1+y^2)}$ (Cauchy), and functional $h(y) = \mathbf{1}_{-10<y<10}$.

- Recall that for Cauchy, $\Pi(0 < X < y) = \arctan(y)/\pi$.
- So, $\mathbf{E}_\pi(h) = \Pi(|X| < 10) = 2\arctan(10)/\pi = 0.93655$.
- Again, not exponentially light tails, so not geometrically ergodic.
- Confidence intervals often miss 0.93655. [file "Rcauchy"]

→ EXAMPLE #4: $\pi(y) = \frac{1}{\pi(1+y^2)}$ (Cauchy), and functional $h(y) = \min(y^2, \text{cutoff}^2)$.

- Take e.g. cutoff = 100. Numerical integration: $\mathbf{E}_\pi(h) \doteq 126.3$.
- Again, not exponentially light tails, so not geometrically ergodic.
- CI usually misses 126.3. (iid MC is better.) [file "Rcauchy2"]

- NOTE: Even when CLT holds, it can be rather unstable, e.g. it requires that chain has converged to Π, so it might underestimate v.
  - Estimate of v is very important! And "varfact" is not always reliable!
  - Repeated runs?
  - Another approach is "batch means", whereby chain is broken into m large "batches", which are assumed to be approximately i.i.d.

TEMPERED MCMC:

- Suppose $\Pi(\cdot)$ is multi-modal, i.e. has distinct "parts".
  - (e.g., $\Pi = \frac{1}{2}N(0,1) + \frac{1}{2}N(20,1)$)
- Usual RWM with $Y_n \sim N(X_{n-1}, 1)$ (say) can explore well within each mode, but how to get from one mode to the other?
- Idea: define a sequence $\Pi_1, \Pi_2, \ldots, \Pi_m$ where $\Pi_1 = \Pi$ ("cold"), and Π_τ is flatter for larger τ ("hot").
  - (e.g. $\Pi_\tau = \frac{1}{2}N(0,\tau^2) + \frac{1}{2}N(20,\tau^2)$; file "Rtempered")
- Then for larger τ, $\Pi_\tau(\cdot)$ is flatter, so much easier to get between modes:

```
maxtemp = 10
pitau = function(x, thetemp) {
  if ((thetemp<1) || (thetemp>maxtemp)) return(0.0)
  else return( 0.5 * dnorm(x, 0, thetemp) +
               0.5 * dnorm(x, 20, thetemp) );
}
tf1 = function(x) {pitau(x,1)}; plot(tf1,-10,30)
tf8 = function(x) {pitau(x,8)}; plot(tf8,-10,30)
```

- Proceed by defining a joint Markov chain (x, τ) on $\mathcal{X} \times \{1, 2, \ldots, m\}$, with stationary distribution $\overline{\Pi}$ defined by $\overline{\Pi}(S \times \{\tau\}) = \frac{1}{m}\Pi_\tau(S)$.
  - (Can also use other weights besides $\frac{1}{m}$.)
  - The Markov chain should have both spatial moves (change x) and temperature moves (change τ). ("Simulated Tempering")
  - e.g. perhaps the chain alternates between:
    (a) propose $x' \sim N(x, 1)$, which is accepted with probability $\min\left(1, \frac{\overline{\pi}(x',\tau)}{\overline{\pi}(x,\tau)}\right) = \min\left(1, \frac{\pi_\tau(x')}{\pi_\tau(x)}\right)$;
    (b) propose $\tau' = \tau \pm 1$ (prob $\frac{1}{2}$ each), which is accepted with probability $\min\left(1, \frac{\overline{\pi}(x,\tau')}{\overline{\pi}(x,\tau)}\right) = \min\left(1, \frac{\pi_{\tau'}(x)}{\pi_\tau(x)}\right)$:

```
xlist = rep(0,M)
X = runif(1,-10,30)  # overdispersed
```

```
for (i in 1:M) {
    Y = X + rnorm(1)  # proposal
    if (runif(1) < pitau(Y,temp) / pitau(X,temp))
        X = Y  # accept new X
    newtemp = sample(c(temp-1,temp+1),1) # proposal
    if (runif(1) < pitau(X,newtemp) / pitau(X,temp))
        temp = newtemp  # accept new temp
    xlist[i] = X;
}
```

- Chain should converge to $\overline{\Pi}$.
- Then, only "count" (and e.g. highlight in red) those samples where $\tau = 1$:

```
print(mean( h(xlist[(templist==1) & ((1:M)>B)]) ))
points((1:M)[templist==1], xlist[templist==1],
                                        col="red")
```

- So how well does it work, for this example?
  - Mixing for just Π, without tempering: terrible! (file "Rtempered" with dotempering=FALSE and temp=1; the small claimed standard errors are completely misleading)
  - However, the mixing is much better for larger τ. (file "Rtempered" with dotempering=FALSE and e.g. temp=8)
  - The above "(a)–(b)" Simulated Tempering algorithm converges fairly well to $\overline{\Pi}$. (file "Rtempered", with dotempering=TRUE)
  - So, conditional on $\tau = 1$, sample is $\approx \Pi$.
  - So, average of those $h(x)$ with $\tau = 1$ gives good estimate of $\mathbf{E}_\pi(h)$.

FINDING THE TEMPERED DENSITIES:

- Usually won't "know" about e.g. $\Pi_\tau = \frac{1}{2}\,N(0,\tau^2) + \frac{1}{2}\,N(20,\tau^2)$.
- Instead, can e.g. let $\pi_\tau(x) = c_\tau\,(\pi(x))^{1/\tau}$. (Sometimes write $\beta = 1/\tau$.)
  - Then $\Pi_1 = \Pi$, and π_τ flatter for larger τ – good.
  - (e.g. if $\pi(x)$ density of $N(\mu,\sigma^2)$, then $c_\tau(\pi(x))^{1/\tau}$ density of $N(\mu,\tau\sigma^2)$.)
  - Then the temperature acceptance probability is:

$$\min\left(1,\ \frac{\pi_{\tau'}(x)}{\pi_\tau(x)}\right) \ = \ \min\left(1,\ \frac{c_{\tau'}}{c_\tau}(\pi(x))^{(1/\tau')-(1/\tau)}\right).$$

  - But this depends on the c_τ, which are usually unknown – bad.
  - e.g. in above example, could try $\pi_\tau(x) = \left(\frac{1}{2}N(0,1;x)+\frac{1}{2}N(20,1;x)\right)^{1/\tau}$.
  - But we do not know the normalising constants c_τ, so this is not a valid algorithm and will not converge! [file "Rtempered2"]
  - What to do?

PARALLEL TEMPERING:

- (a.k.a. replica exchange: Swendsen and Wang, 1986)
- (a.k.a. Metropolis-Coupled MCMC, or MCMCMC: Geyer, 1991)
- Again have a sequence $\Pi_1, \Pi_2, \ldots, \Pi_m$ where $\Pi_1 = \Pi$ ("cold"), and Π_τ is flatter for larger τ ("hot").

- e.g. $\pi_\tau(x) = c_\tau\,(\pi(x))^{1/\tau}$, where τ ranges over $\tau_1 = 1, \tau_2, \tau_3, \ldots, \tau_m$.

- Use state space $\mathcal{X}^m$, with m chains, i.e. one chain for each temperature.
  - So, state at time n is $X_n = (X_{n1}, X_{n2}, \ldots, X_{nm})$, where $X_{n\tau}$ is "at temperature τ".

- Stationary distribution is now $\overline{\Pi} = \Pi_1 \times \Pi_2 \times \ldots \times \Pi_m$, i.e. $\overline{\Pi}(X_1 \in S_1,\ X_2 \in S_2,\ \ldots, X_m \in S_m) = \Pi_1(S_1)\ \Pi_2(S_2)\ \ldots\ \Pi_m(S_m)$.

- Then, can update the chain $X_{n-1,\tau}$ at temperature τ (for each $\tau = 1, 2, \ldots, m$), by proposing e.g. $Y_{n,\tau} \sim N(X_{n-1,\tau},\, 1)$, and accepting with probability $\min\left(1,\ \frac{\pi_\tau(Y_{n,\tau})}{\pi_\tau(X_{n-1,\tau})}\right)$, as usual.

- But can also choose temperatures τ and τ' (e.g., at random, or with $\tau' = \tau + 1$), and propose to "swap" the values $X_{n,\tau}$ and $X_{n,\tau'}$, and accept this with probability $\min\left(1,\ \frac{\pi_\tau(X_{n,\tau'})\,\pi_{\tau'}(X_{n,\tau})}{\pi_\tau(X_{n,\tau})\,\pi_{\tau'}(X_{n,\tau'})}\right)$.
  - Now, normalising constants cancel, e.g. if $\pi_\tau(x) = c_\tau\,(\pi(x))^{1/\tau}$, then acceptance probability is:

$$\min\left(1,\ \frac{c_\tau \pi(X_{n,\tau'})^{1/\tau}\, c_{\tau'} \pi(X_{n,\tau})^{1/\tau'}}{c_\tau \pi(X_{n,\tau})^{1/\tau}\, c_{\tau'} \pi(X_{n,\tau'})^{1/\tau'}}\right) \;=\; \min\left(1,\ \frac{\pi(X_{n,\tau'})^{1/\tau}\, \pi(X_{n,\tau})^{1/\tau'}}{\pi(X_{n,\tau})^{1/\tau}\, \pi(X_{n,\tau'})^{1/\tau'}}\right),$$

    so c_τ and $c_{\tau'}$ are not required.
  - Hence, can set $\pi_\tau(x) = \pi(x)^{1/\tau}$, no problem.

```
pitau = function(x, thetemp) {
  if ((thetemp<1) || (thetemp>maxtemp)) return(0.0)
  else return(pi(x)^(1/thetemp))
}
for (i in 1:M) {
  for (temp in 1:maxtemp) {
    Y = X[temp] + rnorm(1)  # Proposed X[temp]
    if (runif(1) < pitau(Y,temp)/pitau(X[temp],temp)
      X[temp] = Y # Accept
  }
  j = sample(1:(maxtemp-1),1); k=j+1 # Proposed Swap
  if (runif(1) < pitau(X[j],k) * pitau(X[k],j)) /
                 (pitau(X[j],j) * pitau(X[k],k))
    tmpval=X[j]; X[j]=X[k]; X[k]=tmpval # Accept Swap
}
```

- EXAMPLE: again $\Pi = \frac{1}{2}\,N(0,1) + \frac{1}{2}\,N(20,1)$.
  - Now can set $\pi_\tau(x) = \pi(x)^{1/\tau}$, and ignore c_τ.
  - Then run parallel tempering ... works pretty well. [file "Rpara"]

- Optimal choice of τ values? Some partial theory here and here.

END WEEK #8

MONTE CARLO OPTIMISATION – Simulated Annealing:

- General method to find highest mode of π.

- Idea: mode of π is same as mode of a flatter or a more peaked version $\pi_\tau \equiv \pi^{1/\tau}$ as above, for any $\tau > 0$.

- Can this help us?

  - For large τ, MCMC explores a lot; good at beginning of search.
  - For small τ, MCMC narrows in on local mode; good at end of search.

- So, use tempered MCMC, but where $\tau = \tau_n \searrow 0$, so π_{τ_n} becomes more and more concentrated at mode as $n \to \infty$.

- Need to choose $\{\tau_n\}$, the "cooling schedule".
  - e.g. geometric ($\tau_n = \tau_0 \, r^n$ for some $r < 1$):

```
temp = 100; finaltemp = 0.1
tempfactor = (finaltemp/temp)^(1/M)
for (i in 1:M) {
  temp = tempfactor * temp
  Y = X + rnorm(1)
  if (runif(1) < pi(Y,temp) / pi(X,temp))
    X = Y  # accept proposal
}
```

- EXAMPLE: $\Pi_\tau = 0.3\, N(0, \tau^2) + 0.7\, N(20, \tau^2)$. [file "Rsimann"]
  - Highest mode is at 20 (for any τ).
  - If run usual Metropolis algorithm, it will either jump forever between modes (if τ large), or get stuck in one mode or the other with equal probability (if τ small) – bad.
  - But if $\tau_n \searrow 0$ slowly, then can usually find the highest mode (20) – good. (Though sometimes gets stuck near 0.)
  - Try both geometric and linear (better?) cooling ... [file "Rsimann"]

- EXAMPLE with real density powers:
  - Set $\pi_\tau(x) = \Big(0.3\, N(0,1) + 0.7\, N(20,1)\Big)^{1/\tau}$.
  - Need longer run, and smaller final τ.
  - Then it works pretty well. [file "Rsimann2"]

- Choice of cooling schedule:
  - Could instead use e.g. a linear cooling schedule, where $\tau_n = \tau_0 - dn$ for some $d > 0$ (chosen so that $\tau_M \equiv \tau_0 - dM > 0$):

```
tempdiff = (temp-finaltemp)/M
...
  temp = temp - tempdiff
```

  - Or, could choose logarithmic ($\tau_n = \tau_0 / \log(1+n)$). Or ...
  - Advantages? Disadvantages? Best choice?
  - Theorem: If $c \geq \sup \pi$, then simulated annealing with logarithmic cooling $\tau_n = c/\log(1+n)$ will converge to the maximum as $n \to \infty$.
  - Great ... but too slow for practical use.

DIGRESSION – CODE BREAKING:

- Try it out: "decipherdemo". [uses file "decipher.c"]

- Data is the coded message text: $s_1 s_2 s_3 \ldots s_N$, where $s_i \in \mathcal{A} = \{A, B, C, \ldots, Z, \text{space}\}$.

- State space $\mathcal{X}$ is set of all bijections (for now) of $\mathcal{A}$, i.e. one-to-one onto mappings $f : \mathcal{A} \to \mathcal{A}$, subject to $f(\text{space}) = \text{space}$.

- ["substitution cipher"]

- Use a reference text (e.g. "War and Peace") to get matrix $M(x, y) = 1+$ number of times y follows x, for $x, y \in \mathcal{A}$.

- Then for $f \in \mathcal{X}$, let $\pi(f) = \prod_{i=1}^{N-1} M\Big(f(s_i), f(s_{i+1})\Big)$.
  - (Or raise this all to a power, e.g. 0.25.)

- Idea: if $\pi(f)$ is larger, then f leads to pair frequencies which more closely match the reference text, so f is a "better" choice.

- Would like to find the choice of f which maximises $\pi(f)$.

- To do this, run a "Metropolis algorithm" for π:
  - Choose $a, b \in \mathcal{A} \setminus \{\text{space}\}$, uniformly at random.
  - Propose to replace f by g, where $g(a) = f(b)$, $g(b) = f(a)$, and $g(x) = f(x)$ for all $x \neq a, b$.
  - Accept with probability $\min\left(1, \frac{\pi(g)}{\pi(f)}\right)$.

- Easily seen to be an irreducible, reversible Markov chain.

- So, converges (quickly!) to correct answer, breaking the code.

- References: S. Conner (2003), "Simulation and solving substitution codes". P. Diaconis (2008), "The Markov Chain Monte Carlo Revolution".

- We later extended this, to transposition ciphers and more: J. Chen and J.S. Rosenthal (2010), "Decrypting Classical Cipher Text Using Markov Chain Monte Carlo" (*Statistics and Computing* **22(2)**, 397–413, 2011).

DIGRESSION – PATTERN DETECTION:

- Data is an image, given in terms of a grid of pixels (each "on" or "off").

- Want to "find" the face in the image.
  - (Harder for computers than for humans!)

- Java applet: faces.html (See before and after images.)

- Define the face location by a vector θ of various parameters (face center, eye width, nose height, etc.).

- Then define a score function $S(\theta)$ indicating how well the image agrees with having a face in the location corresponding to the parameters θ.

- Then run a "mixed" Monte Carlo search (sometimes updating by small RWM moves, sometimes starting fresh from a random vector) over the entire parameter space, searching for $\text{argmax}_\theta\, S(\theta)$, i.e. for the parameter values which maximise the score function.
  - Keep track of the best θ so far – this allows for greater flexibility in trying different search moves without needing to preserve a stationary distribution.
  - Works pretty well, and fast! ("faces.html" Java applet)
  - For details, see Java applet source code file "faces.java", or the paper J.S. Rosenthal, Optimising Monte Carlo Search Strategies for Automated Pattern Detection. F. E. J. Math. Sci. 2009.

- Here, again, we want to maximise (i.e., optimise) π, not sample from π.

OPTIMAL RWM PROPOSAL SHAPE:

- Consider RWM on $\mathcal{X} = \mathbf{R}^d$, where $Y_n \sim MVN(X_{n-1}, \Sigma)$ for some $d \times d$ proposal covariance matrix Σ.
- What is best choice of Σ?
  - Usually we take $\Sigma = \sigma^2 I_d$ for some $\sigma > 0$, and then choose σ so acceptance rate not too small, not too large (e.g. 0.234).
  - But can we do better?
- Suppose for now that $\Pi = MVN(\mu_0, \Sigma_0)$ for some fixed μ_0 and Σ_0, in dim=5. Try RWM with various proposal distributions [file "Ropt"]:
  - first version: $Y_n \sim MVN(X_{n-1}, I_d)$. ($acc \approx 0.06$; $varfact \gtrapprox 100$)
  - second version: $Y_n \sim MVN(X_{n-1}, 0.1\, I_d)$. ($acc \approx 0.234$; $varfact \gtrapprox 100$)
  - third version: $Y_n \sim MVN(X_{n-1}, \Sigma_0)$. ($acc \approx 0.31$; $varfact \approx 15$)
  - fourth version: $Y_n \sim MVN(X_{n-1}, 1.4\,\Sigma_0)$. ($acc \approx 0.234$; $varfact \approx 7$)
- Or in dim=20 (file "Ropt2", which uses the auxiliary file "Rtarg20"):
  - $Y_n \sim MVN(X_{n-1}, 0.025\, I_d)$. ($acc \approx 0.234$; $varfact \gtrapprox 400$)
  - $Y_n \sim MVN(X_{n-1}, 0.283\, \Sigma_0)$. ($acc \approx 0.234$; $varfact \approx 50$)
- Conclusion: acceptance rates near 0.234 are better.
- But also, proposals shaped like the target (i.e., with covariance matrices proportional to the target covariance matrix) are better.
  - Indeed, best is when proposal covariance $= ((2.38)^2/d)\Sigma_0$.
  - Seems to lead to $acc \approx 0.234$ and faster convergence.
  - This has been proven under strong assumptions, e.g. targets which are orthogonal transformations of independent components (Roberts et al., Ann Appl Prob 1997; Roberts and Rosenthal, Stat Sci 2001 ; Bédard, Ann Appl Prob 2007).
  - Seems to be "approximately" true for most unimodal targets ...
- Problem: Σ_0 would usually be unknown; then what?
  - Can perhaps "adapt" ...

END WEEK #9

(Student presentations ...)

ADAPTIVE MCMC:

- Recall: RWM optimal proposal covariance is $((2.38)^2/d)\Sigma_0$.
- What if target covariance Σ_0 is unknown??
- Can estimate Σ_0 based on run so far, to get empirical covariance Σ_n.
- Then update proposal covariance "on the fly".
- "Learn as you go": see e.g. the Javascript simulation.
- For Adaptive MCMC, could use proposal $Y_n \sim MVN(X_{n-1}, ((2.38)^2/d)\Sigma_n)$.
  - Hope that for large n, $\Sigma_n \approx \Sigma_0$, so proposals "nearly" optimal.
  - (Usually also add ϵI_d to proposal covariance, to improve stability, e.g. $\epsilon = 0.05$.)

- Try R version, for the same MVN example as in Ropt [file "Radapt"]:
  - Need much longer burn-in, e.g. $B = 20,000$, for adaption to work.
  - Get varfact of last 4000 iterations of about 18 . . . "competitive" with Ropt optimal . . .
  - The longer the run, the more benefit from adaptation.
  - Can also compute "slow-down factor", $s_n \equiv d\left(\sum_{i=1}^d \lambda_{in}^{-2} / (\sum_{i=1}^d \lambda_{in}^{-1})^2\right)$, where $\{\lambda_{in}\}$ eigenvals of $\Sigma_n^{1/2}\,\Sigma_0^{-1/2}$. Starts large, should converge to 1. [Motivation: if $\Sigma_n = \Sigma_0$, then $\lambda_{in} \equiv 1$, so $s_n = d(d/d^2) \equiv 1$.] See Roberts and Rosenthal, Examples of Adaptive MCMC, JCGS 2009.
- Higher dimensions: figure in file "RplotAMx200.png" (dim=200).
  - Works well, but takes many iterations before the adaption is helpful.
- Another approach: Adaptive Independence Sampler
  - Use information from the run so far, to find a good Independence Sampler proposal $q(y)$ (e.g. an appropriate mixture of normal densities) which approximates $\pi(y)$.
  - If so, this leads to a nearly-optimal Independence Sampler (as discussed previously).
  - For more details, see: Giordani & Kohn, JCGS 2010

CONVERGENCE OF ADAPTIVE MCMC:

- Is Adaptive MCMC a valid algorithm?
  - Will it necessarily converge to Π??
  - Not in general! See e.g. the Javascript simulation.
  - Algorithm is now non-Markovian; doesn't preserve stationarity.
- However, adaptive MCMC is still guaranteed to converge to Π under various additional conditions.
- For example, it suffices (see Roberts & Rosenthal, "Coupling and Convergence of Adaptive MCMC" (J. Appl. Prob. 2007)) that the adaption satisfies:
  - (a) Diminishing Adaptation: Adapt less and less as the algorithm proceeds. Formally, $\sup_{x\in\mathcal{X}} \|P_{\Gamma_{n+1}}(x,\cdot) - P_{\Gamma_n}(x,\cdot)\| \to 0$ in prob. [Can always be made to hold, since adaption is user controlled.]
  - (b) Containment: For all $\epsilon > 0$, the time to converge to within ϵ of stationary from $x = X_n$, if fix $\gamma = \Gamma_n$, remain bounded in probability as $n \to \infty$. [Technical condition, to avoid "escape to infinity". Holds if e.g. the state space and adaption spaces are both compact, etc. And always seems to hold in practice.]
  - (This also guarantees WLLN for bounded functionals. Various other results about LLN / CLT under stronger assumptions.)
  - There are various "checkable" sufficient conditions which guarantee Containment, e.g. Y. Bai, G.O. Roberts, and J.S. Rosenthal, Adv. Appl. Stat. 2011 and Craiu, Gray, Latusynski, Madras, Roberts, and Rosenthal, Ann. Appl. Prob. 2015 and J.S. Rosenthal and J. Yang, Ergodicity of Discontinuous Adaptive MCMC Algorithms, MCAP 2018.
- So, some "reasonable" theory, but you have to be careful.

TRANSDIMENSIONAL MCMC:

NOTE: I did not have time to discuss this in class, but discussed it afterwards with a few students; I include it here for your interest and information.

- (a.k.a. "reversible-jump MCMC": Green, Biometrika 1995)
- What if the state space is a union of parts of different dimension?
  - Can we still apply Metropolis-Hastings then??
- (EXAMPLE: autoregressive process: suppose $Y_n = a_1 Y_{n-1} + a_2 Y_{n-2} + \ldots + a_k Y_{n-k}$, but we don't know what k should be.)
- OUR EXAMPLE: suppose $\{y_j\}_{j=1}^J$ are known data which are assumed to come from a mixture distribution: $\frac{1}{k}(N(a_1,1)+N(a_2,1)+\ldots+N(a_k,1))$.
- Want to estimate the unknown $k, a_1, \ldots, a_k$.
  - Here the number of parameters is also unknown, i.e. the dimension is unknown and variable, which makes MCMC more challenging!
- The state space is $\mathcal{X} = \{(k,a) : k \in \mathbf{N},\ a \in \mathbf{R}^k\}$.
- Prior distributions: $k-1 \sim$ Poisson(2), and $a|k \sim MVN(0, I_k)$ (say).
- Define a reference measure λ by: $\lambda(\{k\} \times A) = \lambda_k(A)$ for $k \in \mathbf{N}$ and (measurable) $A \subseteq \mathbf{R}^k$, where λ_k is Lebesgue measure on $\mathbf{R}^k$.
  - i.e., $\lambda = \delta_1 \times \lambda_1 + \delta_2 \times \lambda_2 + \delta_3 \times \lambda_3 + \ldots$
- Then in our mixture example, posterior density (with respect to λ) is:

$$\pi(k,a) \;=\; C\,\frac{e^{-2}2^{k-1}}{(k-1)!}(2\pi)^{-k/2}\exp\left(-\frac{1}{2}\sum_{i=1}^k a_i^2\right)(2\pi)^{-J/2}\prod_{j=1}^J\left(\sum_{i=1}^k \frac{1}{k}\exp\left(-\frac{1}{2}(y_j-a_i)^2\right)\right).$$

- So, on a log scale,

$$\log\pi(k,a) \;=\; \log C + \log\frac{e^{-2}2^{k-1}}{(k-1)!} - \frac{k}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^k a_i^2 - \frac{J}{2}\log(2\pi) +$$

$$\sum_{j=1}^J \log\left(\sum_{i=1}^k \frac{1}{k}\exp\left(-\frac{1}{2}(y_j-a_i)^2\right)\right).$$

  (Can ignore $\log C$ and $\frac{J}{2}\log(2\pi)$, but not $\frac{k}{2}\log(2\pi)$.)
- How to "explore" this posterior distribution??
- For fixed k, can move around $\mathbf{R}^k$ in usual way with RWM (say).
- But how to change k?
- Can propose to replace k with, say, $k' = k \pm 1$ (prob $\frac{1}{2}$ each).
- Then have to correspondingly change a. One possibility:
  - If $k' = k+1$, then $a' = (a_1, \ldots, a_k, Z)$ where $Z \sim N(0,1)$ ("elongate").
  - If $k' = k-1$, then $a' = (a_1, \ldots, a_{k-1})$ ("truncate").
- Then accept with usual probability, $\min\left(1,\ \frac{\pi(k',a')\,q\big((k',a'),(k,a)\big)}{\pi(k,a)\,q\big((k,a),(k',a')\big)}\right)$.
  - Here if $k' = k+1$, then $q((k',a'),(k,a)) = \frac{1}{2}$, while $q((k,a),(k',a')) = \frac{1}{2}\frac{1}{\sqrt{2\pi}}e^{-(a'_{k'})^2/2}$.

- Or, if $k' = k-1$, then $q((k,a),(k',a')) = \frac{1}{2}$, while $q((k',a'),(k,a)) = \frac{1}{2}\frac{1}{\sqrt{2\pi}}e^{-(a_k)^2/2}$.

- Seems to work okay; final k usually between 5 and 9 ... [file "Rtrans"]
- ALTERNATIVE method for the "correspondingly change a" step:
  - If $k' = k+1$, then $a' = (a_1, \ldots, a_{k-1}, a_k - Z, a_k + Z)$ where $Z \sim N(0,1)$ ("split").
  - If $k' = k-1$, then $a' = (a_1, \ldots, a_{k-2}, \frac{1}{2}(a_{k-1} + a_k))$ ("merge").
  - What about the densities $q((k',a'),(k,a))$?
  - Well, if $k' = k+1$, then $q((k',a'),(k,a)) = \frac{1}{2}$, while roughly speaking,

$$q((k,a),(k',a')) \;=\; \frac{1}{2}\frac{1}{\sqrt{2\pi}}e^{-z^2/2} \;=\; \frac{1}{2}\frac{1}{\sqrt{2\pi}}e^{-(\frac{1}{2}(a'_{k'}-a'_k))^2/2}\,.$$

  - One subtle additional point: The map $(a, Z) \mapsto a' = (a_1, \ldots, a_{k-1}, a_k - Z, a_k + Z)$ has "Jacobian" term:

$$\det\left(\tfrac{\partial a'}{\partial(a,Z)}\right) \;=\; \det\begin{pmatrix} I_{k-1} & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{pmatrix} \;=\; 1-(-1) \;=\; 2\,,$$

    i.e. the split moves "spread out" the mass by a factor of 2.
  - So by Change-of-Variable Thm, actually

$$q((k,a),(k',a')) \;=\; \frac{1}{2}\frac{1}{\sqrt{2\pi}}e^{-(\frac{1}{2}(a'_{k'}-a'_k))^2/2} \;\Big/\; 2\,.$$

  - Similarly, if $k' = k-1$, then $q((k,a),(k',a')) = \frac{1}{2}$, while

$$q((k',a'),(k,a)) \;=\; \frac{1}{2}\frac{1}{\sqrt{2\pi}}e^{-(\frac{1}{2}(a_k-a_{k'}))^2/2} \;\Big/\; 2\,.$$

  - Algorithm still seems to work okay ... [file "Rtrans2"]
- For more complicated transformations, need to include more complicated "Jacobian" term (but above it equals 1 or 2).
- Check: if we start the algorithms with, say, $k = 24$, then they don't manage to reduce k enough!
  - They might be trying to remove the "wrong" a_i.
- So, try another MODIFICATION, this time where any coordinate can be added/removed, not just the last one.
  - While we're at it, change "new a_i distribution" from $Z \sim N(0,1)$ to $Z \sim \text{Uniform}(-20, 30)$, with corresponding change to the $q((k,a),(k',a'))$ formulae.
  - file "Rtrans3" – now works well even if started with $k = 24$.
  - Seems to settle on $k = 6$ regardless of starting value.
  - This seems to indicate rapid mixing – good!

END WEEK #10

(More student presentations ...)

- FINAL SUMMARY: Monte Carlo can be used for nearly everything!
  - Good luck with the final project, and with all the rest of your studies.