Markov chain Monte Carlo[1]

- Historical background
- Gibbs sampler
- Metropolis-Hastings algorithms

[1] Based on Gamerman and Lopes (2007) *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, Chapman&Hall/CRC.

Dongarra and Sullivan (2000) Guest Editors' Introduction: The Top 10 Algorithms, *Computing in Science and Engineering*, **2**, 22-23.

The top 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century (in chronological order):

1. Metropolis Algorithm for Monte Carlo
2. Simplex Method for Linear Programming
3. Krylov Subspace Iteration Methods
4. The Decompositional Approach to Matrix Computations
5. The Fortran Optimizing Compiler
6. QR Algorithm for Computing Eigenvalues
7. Quicksort Algorithm for Sorting
8. Fast Fourier Transform
9. Integer Relation Detection
10. Fast Multipole Method

Andrieu, de Freitas, Doucet and Jordan (2003) An Introduction to MCMC for machine learning, *Machine Learning*, **50**, 5-43.

- "While convalescing from an illness in 1946, Stan Ulam was playing solitaire. It, then, occurred to him to try to compute the chances that a particular solitaire laid out 52 cards would come out successfully.
- After attempting exhaustive combinatorial calculations, he decided to go for the more practical approach of laying out several solitaires at random and then observing and counting the number of successful plays.
- This idea of selecting a statistical sample to approximate a hard combinatorial problem by a much simpler problem is the heart of modern Monte Carlo simulation."

Eckhard (1987) Stan Ulam, John Von Neumann and the Monte Carlo method. *Los Alamos Science*, **15**, 131-136.
Metropolis and Ulam (1949) The Monte Carlo method. *Journal of the American Statistical Association*, **44**, 335-341;

1940s and 1950s

Stan Ulam soon realized that computers could be used in this fashion to answer questions of neutron diffusion and mathematical physics;

He contacted John Von Neumann and they developed many Monte Carlo algorithms (importance sampling, rejection sampling, etc);

In the 1940s Nick Metropolis and Klari Von Neumann designed new controls for the state-of-the-art computer (ENIAC);

Metropolis and Ulam (1949) The Monte Carlo method. *Journal of the American Statistical Association*, **44**, 335-341;

Metropolis, Rosenbluth, Rosenbluth, Teller and Teller (1953) Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087-1091.

1970s

Hastings and his student Peskun showed that Metropolis and the more general Metropolis-Hastings algorithm are particular instances of a larger family of algorithms.

Hastings (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97-109.

Peskun (1973) Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, **60**, 607-612.

1980s

Geman and Geman (1984)
Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 721-741.

Pearl (1987)
Evidential reasoning using stochastic simulation. *Artificial Intelligence*, **32**, 245-257.

1990s

Tanner and Wong (1987)
The calculation of posterior distributions by data augmentation.
Journal of the American Statistical Association, **82**, 528-550.

Gelfand and Smith (1990)
Sampling-based approaches to calculating marginal densities.
Journal of the American Statistical Association, **85**, 398-409.

Gibbs sampler

A sequence $\{\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \dots\}$ is drawn from a Markov chain whose *limiting equilibrium distribution* is the posterior distribution, $\pi(\theta)$, and whose transition kernel is the product of the full conditional distributions:

Algorithm

1. $\theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \dots, \theta_p^{(0)})$
2. $\theta^{(j)}$ is sampled as follows:

$$\begin{array}{rcl}
\theta_1^{(j)} & \sim & \pi(\theta_1 | \theta_2^{(j-1)}, \dots, \theta_p^{(j-1)}) \\
\theta_2^{(j)} & \sim & \pi(\theta_2 | \theta_1^{(j)}, \theta_3^{(j-1)}, \dots, \theta_p^{(j-1)}) \\
 & \vdots & \\
\theta_p^{(j)} & \sim & \pi(\theta_p | \theta_1^{(j)}, \dots, \theta_{p-1}^{(j)})
\end{array}$$

Example 0. Simple linear regression

For $i = 1, \ldots, n$, y_i is linearly related to x_i, ie.

$$y_i \sim N(\alpha + \beta x_i, \sigma^2)$$

with y's independent conditionally on $\theta = (\alpha, \beta, \sigma^2)$.

Prior distribution:

$$\begin{array}{rcl} \alpha & \sim & N(\alpha_0, \tau_\alpha^2) \\ \beta & \sim & N(\beta_0, \tau_\beta^2) \\ \sigma^2 & \sim & IG(\nu_0/2, \nu_0 \sigma_0^2/2) \end{array}$$

No analytical solution to $E(g(\theta)|x, y)$

Easy to derive full conditionals $\Rightarrow$ Gibbs sampler!

Full conditionals

- $[\alpha] \sim N(m, C)$

$$\begin{aligned} m &= C\left(\tau_\alpha^{-2}\alpha_0 + \sigma^{-2}\sum_{i=1}^{n}(y_i - \beta x_i)\right) \\ C^{-1} &= \tau_\alpha^{-2} + \sigma^{-2} n \end{aligned}$$

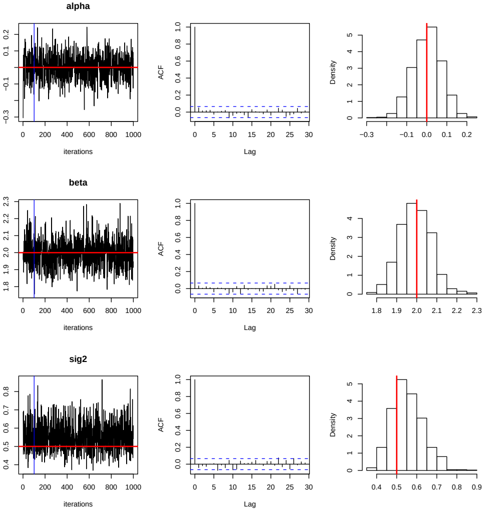
- $[\beta] \sim N(m, C)$

$$\begin{aligned} m &= C\left(\tau_\beta^{-2}\beta_0 + \sigma^{-2}\sum_{i=1}^{n}(y_i - \alpha)x_i\right) \\ C^{-1} &= \tau_\beta^{-2} + \sigma^{-2}\sum_{i=1}^{n} x_i^2 \end{aligned}$$

- $[\sigma^2] \sim IG\left(\nu_1/2, \nu_1\sigma_1^2/2\right)$

$$\begin{aligned} \nu_1 &= \nu_0 + n \\ \nu_1\sigma_1^2 &= \nu_0\sigma_0^2 + \sum_{i=1}^{n}(y_i - \alpha - \beta x_i)^2 \end{aligned}$$

```
# Simulating the data
set.seed(1244)
n=100;alpha=0;beta=2;sig2=0.5;true=c(alpha,beta,sig2)
x=rnorm(n)
y=rnorm(n,alpha+beta*x,sqrt(sig2))
# Prior hyperparameters
alpha0=0;tau2a=10;beta0=0;tau2b=10;nu0=3;s02=1;nu0s02=nu0*s02
# Setting up starting values
alpha=0;beta=0;sig2=1
# Gibbs sampler
M = 1000
draws = matrix(0,M,3)
for (i in 1:M){
  var   = 1/(1/tau2a+n/sig2)
  mean  = var*(sum(y-beta*x)/sig2+alpha0/tau2a)
  alpha = rnorm(1,mean,sqrt(var))
  var   = 1/(1/tau2b+sum(x^2)/sig2)
  mean  = var*(sum((y-alpha)*x)/sig2+beta0/tau2b)
  beta  = rnorm(1,mean,sqrt(var))
  sig2  = 1/rgamma(1,(nu0+n)/2,(nu0s02+sum((y-alpha-beta*x)^2)/2))
  draws[i,] = c(alpha,beta,sig2)
}
# Markov chains + marginal posterior
names = c("alpha","beta","sig2")
ind = 101:M
par(mfrow=c(3,3))
for (i in 1:3){
  ts.plot(draws[,i],xlab="iterations",ylab="",main=names[i])
  abline(v=ind[1],col=4)
  abline(h=true[i],col=2,lwd=2)
  acf(draws[ind,i],main="")
  hist(draws[ind,i],prob=T,main="",xlab="")
  abline(v=true[i],col=2,lwd=2)
}
```



Simulation: n=100, α=0, β=2, $\sigma^2 = 0.5$. Initial values: $\alpha^{(0)} = 0$, $\beta^{(0)} = 0$ and $\sigma^{2(0)} = 1$. 1000 draws, 100 burn-in.

Example i. Poisson with a break

$y_1, \ldots, y_n$ is a sample from a Poisson distribution.

There is a suspicion of a change point m along the observation process.

Given m, the observation distributions are

$$\begin{aligned} y_i|\lambda &\sim Poi(\lambda), \quad i = 1, \ldots, m \\ y_i|\phi &\sim Poi(\phi), \quad i = m+1, \ldots, n \end{aligned}$$

Independent prior distributions

$$\begin{aligned} \lambda &\sim G(\alpha, \beta) \\ \phi &\sim G(\gamma, \delta) \\ m &\sim U\{1, \ldots, n\} \end{aligned}$$

where α, β, γ and δ are known constants.

The posterior density is

$$\begin{aligned}
\pi(\lambda, \phi, m) &\propto f(y_1, \dots, y_n|\lambda, \phi, m) p(\lambda, \phi, m) \\
&= \prod_{i=1}^{m} f_P(y_i; \lambda) \prod_{i=m+1}^{n} f_P(y_i; \phi) \\
&\times f_G(\lambda; \alpha, \beta) f_G(\phi; \gamma, \delta) \frac{1}{n} \\
&\propto \lambda^{\alpha+s_m-1} e^{-(\beta+m)\lambda} \\
&\times \phi^{\gamma+s_n-s_m-1} e^{-(\delta+n-m)\phi}
\end{aligned}$$

where $s_l = \sum_{i=1}^{l} y_i$ for $l = 1, \dots, n$.

It becomes simple to obtain the full conditional densities

$$\begin{aligned}
\pi_\lambda(\lambda) &= G\left(\alpha + s_m, \beta + m\right), \\
\pi_\phi(\phi) &= G\left(\gamma + s_n - s_m, \delta + n - m\right), \\
\pi_m(m) &= \frac{\lambda^{\alpha+s_m-1} e^{-(\beta+m)\lambda} \phi^{\gamma+s_n-s_m-1} e^{-(\delta+n-m)\phi}}{\sum_{l=1}^{n} \lambda^{\alpha+s_l-1} e^{-(\beta+l)\lambda} \phi^{\gamma+s_n-s_l-1} e^{-(\delta+n-l)\phi}},
\end{aligned}$$

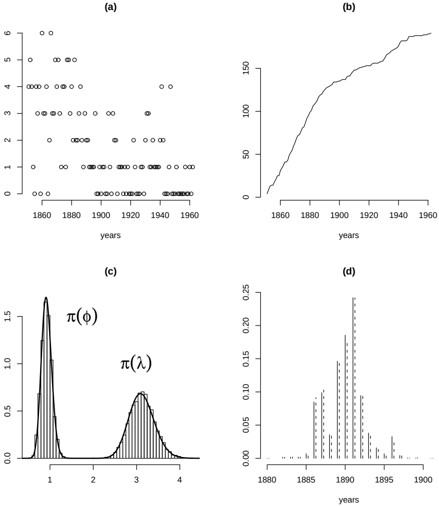
for $m = 1, \dots, n$.

This model was applied to the $n = 112$ observed counts of coal mining disasters in Great Britain by year from 1851 to 1962.
The Gibbs sampler run: 5000 iterations
Starting point: $m^{(0)} = 1891$
Hyperparameters: $\alpha = \beta = \gamma = \delta = 0.001$
Exact posterior quantities.

Par.	Mean	Var	95% C.I.
α	3.120	0.280	(2.571,3.719)
β	0.923	0.113	(0.684,0.963)
m	1890	2.423	(1886,1895)

Approximate posterior quantities.

Par.	Mean	Var	95% C.I.
α	3.131	0.290	(2.582,3.733)
β	0.922	0.118	(0.703,1.167)
m	1890	2.447	(1886,1896)



(a) counts of coal mining disasters from 1851 to 1962, (b) cumulative counts, (c) true and approximations of $\pi(\lambda)$ and $\pi(\phi)$, (d) true and approximations of $\pi(m)$.

Metropolis-Hastings

A sequence $\{\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \dots\}$ is drawn from a Markov chain whose *limiting equilibrium distribution* is the posterior distribution, $\pi(\theta)$.

Algorithm

1. Initial value: $\theta^{(0)}$
2. Proposed move: $\theta^* \sim q(\theta^* | \theta^{(i-1)})$
3. Acceptance scheme:

$$\theta^{(i)} = \begin{cases} \theta^* & \text{com prob. } \alpha \\ \theta^{(i-1)} & \text{com prob. } 1-\alpha \end{cases}$$

where

$$\alpha = \min\left\{1, \frac{\pi(\theta^*)}{\pi(\theta^{(i-1)})} \frac{q(\theta^* | \theta^{(i-1)})}{q(\theta^{(i-1)} | \theta^*)}\right\}$$

Special cases

1. Random walk chains: $q(\theta|\theta^*) = q(|\theta - \theta^*|)$

$$\alpha = \min\left\{1, \frac{\pi(\theta^*)}{\pi(\theta)}\right\}$$

   such that a value θ^* with density $\pi(\theta^*)$ greater than $\pi(\theta)$ is automatically accepted.

2. Independence chains: $q(\theta|\theta^*) = q(\theta)$

$$\alpha = \min\left\{1, \frac{\omega(\theta^*)}{\omega(\theta)}\right\}$$

   where $\omega(\theta^*) = \pi(\theta^*)/q(\theta^*)$.

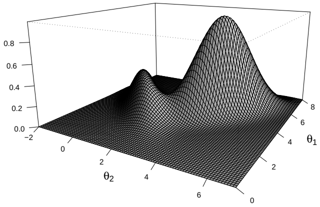
Example ii.

In this example the target distribution is a two-component mixture of bivariate normal densities, ie:

$$\pi(\theta) = 0.7 f_N(\theta; \mu_1, \Sigma_1) + 0.3 f_N(\theta; \mu_2, \Sigma_2).$$

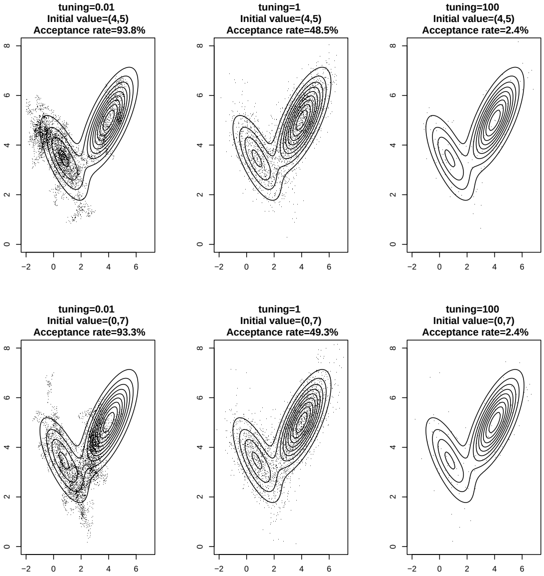
where $\mu_1' = (4, 5)$, $\mu_2' = (0.7, 3.5)$,

$$\Sigma_1 = \begin{pmatrix} 1.0 & 0.7 \\ 0.7 & 1.0 \end{pmatrix} \text{ and } \Sigma_2 = \begin{pmatrix} 1.0 & -0.7 \\ -0.7 & 1.0 \end{pmatrix} .$$



Random walk Metropolis

$q(\theta, \phi) = f_N(\phi; \theta, \nu I_2)$ and ν =tuning.



tuning=0.01
Initial value=(4,5)
Acceptance rate=93.8%

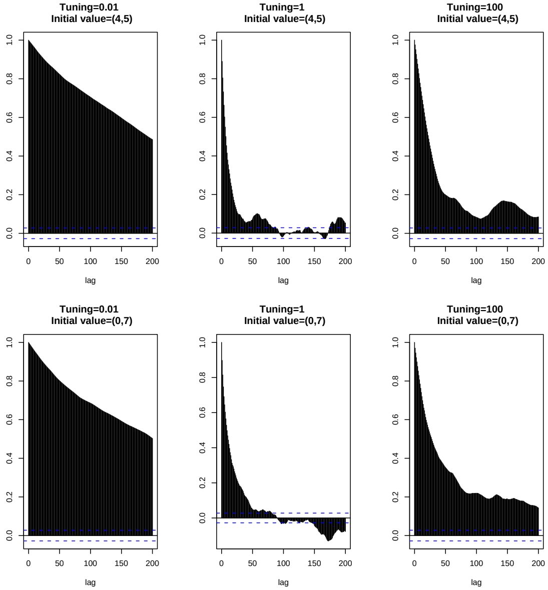
tuning=1
Initial value=(4,5)
Acceptance rate=48.5%

tuning=100
Initial value=(4,5)
Acceptance rate=2.4%

tuning=0.01
Initial value=(0,7)
Acceptance rate=93.3%

tuning=1
Initial value=(0,7)
Acceptance rate=49.3%

tuning=100
Initial value=(0,7)
Acceptance rate=2.4%



Tuning=0.01
Initial value=(4,5)

Tuning=1
Initial value=(4,5)

Tuning=100
Initial value=(4,5)

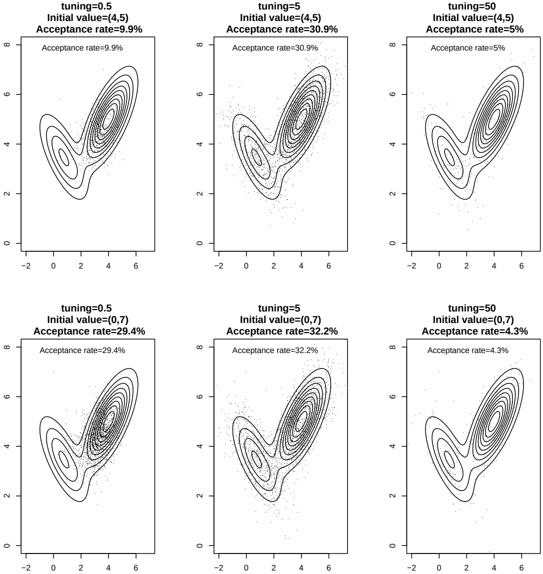
Tuning=0.01
Initial value=(0,7)

Tuning=1
Initial value=(0,7)

Tuning=100
Initial value=(0,7)

Independent Metropolis

$q(\theta, \phi) = f_N(\phi; \mu_3, \nu I_2)$ and $\mu_3 = (3.01, 4.55)'$.



tuning=0.5
Initial value=(4,5)
Acceptance rate=9.9%

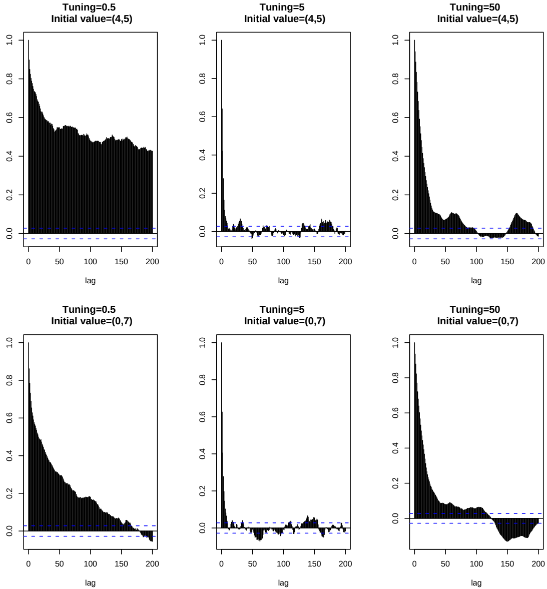
tuning=5
Initial value=(4,5)
Acceptance rate=30.9%

tuning=50
Initial value=(4,5)
Acceptance rate=5%

tuning=0.5
Initial value=(0,7)
Acceptance rate=29.4%

tuning=5
Initial value=(0,7)
Acceptance rate=32.2%

tuning=50
Initial value=(0,7)
Acceptance rate=4.3%


Tuning=0.5
Initial value=(4,5)
Tuning=5
Initial value=(4,5)
Tuning=50
Initial value=(4,5)
Tuning=0.5
Initial value=(0,7)
Tuning=5
Initial value=(0,7)
Tuning=50
Initial value=(0,7)
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
lag

Example iii. Simple t-Student regression

For $i = 1, \dots, n$, y_i is linearly related to x_i, ie.

$$y_i \sim t_\nu(\alpha + \beta x_i, \sigma^2)$$

with y's independent conditionally on α, β and σ^2.

Let us keep ν, σ^2, for simplicity, so $\theta = (\alpha, \beta)$.

Prior distribution:

$$\begin{array}{rcl} \alpha & \sim & N(\alpha_0, \tau_\alpha^2) \\ \beta & \sim & N(\beta_0, \tau_\beta^2) \end{array}$$

No analytical solution to $E(g(\theta)|x, y)$

Full conditionals of no known form!

Let us try a simple random walk Metropolis.

Random walk proposal

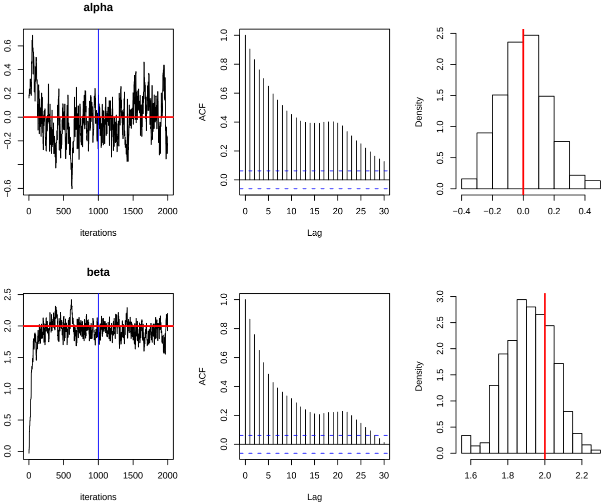
For a given state $\theta^{(i)} = (\alpha, \beta)^{(i)}$, sample

$$\begin{array}{rcl} \alpha^* & \sim & N(\alpha^{(i)}, \nu_\alpha) \\ \phi^* & \sim & N(\phi^{(i)}, \nu_\beta) \end{array}$$

Then, $\theta^{(i+1)} = \theta^*$ with probability α:

$$\alpha = \min\left\{1, \frac{p(\theta^*)p(y|\theta^*)}{p(\theta^{(i)})p(y|\theta^{(i)})}\right\}$$

```
# Simulating the data
set.seed(1244)
n=100;nu=4;alpha=0;beta=2;sig=1;true=c(alpha,beta)
x=rnorm(n,1,1)
y=alpha+beta*x+sig*rt(n,nu)
# Prior hyperparameters
alpha0=0;taua=10;beta0=0;taub=10
# Setting up starting values
a=0;b=0
# Random walk Metropolis
sda=0.1;sdb=0.1
M = 2000
draws = matrix(0,M,2)
for (i in 1:M){
  a1 = rnorm(1,a,sda)
  b1 = rnorm(1,b,sdb)
  num = prod(dt((y-a1-b1*x)/sig,nu))*dnorm(a1,alpha0,taua)*dnorm(b1,beta0,taub)
  den = prod(dt((y-a-b*x)/sig,nu))*dnorm(a,alpha0,taua)*dnorm(b,beta0,taub)
  acc = min(1,num/den)
  u = runif(1)
  if (u<acc){a=a1;b=b1}
  draws[i,] = c(a,b)
}
# Markov chains + marginal posterior
names = c("alpha","beta")
ind = 1001:M
par(mfrow=c(2,3))
for (i in 1:2){
  ts.plot(draws[,i],xlab="iterations",ylab="",main=names[i])
  abline(v=ind[1],col=4)
  abline(h=true[i],col=2,lwd=2)
  acf(draws[ind,i],main="")
  hist(draws[ind,i],prob=T,main="",xlab="")
  abline(v=true[i],col=2,lwd=2)
}
```



Simulation: n=100, $\nu = 4$, α=0, β=2, $\sigma^2 = 1$. Initial values: $\alpha^{(0)} = 0$ and $\beta^{(0)} = 0$. *Random walk Metropolis:* $\nu_\alpha = \nu_\beta = 0.01$, 1000 draws, 1000 burn-in.

Independent Metropolis-Hastings proposal

Sample θ^* from

$$q(\theta) = f_N(\theta, \hat{\theta}, V)$$

where $V = \hat{\sigma}^2(X'X)^{-1}$, $X = (1, x)$, $\hat{\theta} = (X'X)^{-1}X'y$ and $\hat{\sigma}^2 = n^{-1}\sum_{i=1}^n (y_i - \hat{\alpha} - \hat{\beta}x_i)^2$.

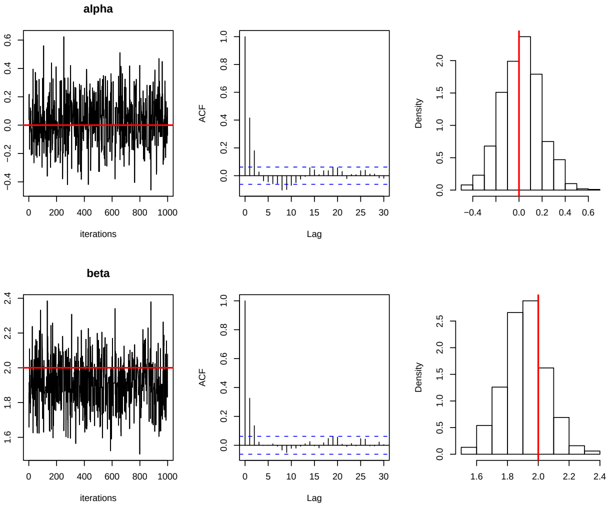
Set

$$\theta^{(i+1)} = \begin{cases} \theta^* & \text{with probability } \alpha \\ \theta^{(i)} & \text{with probability } 1 - \alpha \end{cases}$$

where

$$\alpha = \min\left\{1, \frac{p(\theta^*)p(y|\theta^*)}{p(\theta^{(i)})p(y|\theta^{(i)})}\frac{q(\theta^{(i)})}{q(\theta^*)}\right\}.$$

```
# Simulating the data
set.seed(1244)
n=100;nu=4;theta=c(0,2);sig=1
X = cbind(1,rnorm(n,1,1))
y=X%*%theta+sig*rt(n,nu)
# Prior hyperparameters
theta0=c(0,0);V0=c(10,10)
# Setting up starting values and independent Metropolis-Hastings proposal
reg   = lm(y~X-1)
s2hat = mean(reg$res^2)
thhat = c(solve(t(X)%*%X)%*%t(X)%*%y)
V     = s2hat*solve(t(X)%*%X)
th    = thhat
# MCMC setup
burn=1000;step=1;niter=1000;M=burn+niter*step
draws = matrix(0,M,2)
for (i in 1:M){
  th1 = c(rmvnorm(1,thhat,V))
  num = sum(dt((y-X%*%th1)/sig,nu,log=T))+sum(dnorm(th1,theta0,sqrt(V0),log=T))
  den = sum(dt((y-X%*%th)/sig,nu,log=T))+sum(dnorm(th,theta0,sqrt(V0),log=T))
  num = num-dmvnorm(th1,thhat,V,log=T)
  den = den-dmvnorm(th,thhat,V,log=T)
  if (log(runif(1))<min(0,num-den)){th=th1}
  draws[i,] = th
}
# Markov chains + marginal posterior
names = c("alpha","beta")
ind = seq((burn+1),M,by=step)
par(mfrow=c(2,3))
for (i in 1:2){
  ts.plot(draws[ind,i],xlab="iterations",ylab="",main=names[i])
  abline(h=true[i],col=2,lwd=2)
  acf(draws[ind,i],main="")
  hist(draws[ind,i],prob=T,main="",xlab="")
  abline(v=true[i],col=2,lwd=2)
}
```



Simulation: n=100, $\nu = 4$, α=0, β=2, $\sigma^2 = 1$. Initial values: $\alpha^{(0)} = 0$ and $\beta^{(0)} = 0$. *Independent Metropolis:* 1000 draws, 1000 burn-in.

References

Andrieu, de Freitas, Doucet and Jordan (2003) An Introduction to MCMC for machine learning, *Machine Learning*, **50**, 5-43.

Dongarra and Sullivan (2000) Guest Editors' Introduction: The Top 10 Algorithms, *Computing in Science and Engineering*, **2**, 22-23.

Eckhard (1987) Stan Ulam, John Von Neumann and the Monte Carlo method. *Los Alamos Science*, **15**, 131-136.

Gelfand, A. E. and Smith, A. F. M. (1990) Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, **85**, 398-409.

Geman and Geman (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 721-741.

Hastings (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97-109.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953) Equation of state calculations by fast computing machine. *Journal of Chemical Physics*, **21**, 1087-91.

Metropolis and Ulam (1949) The Monte Carlo method. *Journal of the American Statistical Association*, **44**, 335-341.

Pearl (1987) Evidential reasoning using stochastic simulation. *Artificial Intelligence*, **32**, 245-257.

Peskun (1973) Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, **60**, 607-612.

Tanner and Wong (1987) The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, **82**, 528-550.