Aims

- Markov Chain
- Differences between MC and iterative simulations (IS)
- Metropolis algorithm (MA)
  - Bivariate unit normal density with bivariate normal jumping kernel
  - Theoretical concerns about the MA
- Metropolis Hastings algorithm (MHA)
- Gibbs Sampler (GS)
- Metropolis within Gibbs
- Assessing convergence
- Adaptive simulation algorithm

Markov chain simulation

- IDEA: to simulate a random walk in the space of θ which converge to a stationary distribution that is the posterior distribution $p(\theta \mid y)$
- KEY: to create a Markov process whose stationary distribution is $p(\theta \mid y)$ and run the simulation long enough that the distribution of the current draws is close enough to the target
- the key of the MC's success is that the approximate distributions are improved at each step in the simulation, in the sense of converging to the target distribution, whereas distributions used in the importance sampling remains the same

Differences between MC and IS simulations

- In MC, samples are drawn sequentially, with the distribution of the sampled draws depending on the last value drawn
- The approximate distributions are improved at each step in the simulation, in the sense of converging to the target distribution, whereas distributions used in IS simulations remain the same
- In MC, we draw θ^t from a transition distribution

$$\theta^t \sim T_t(\theta^t \mid \theta^{t-1})$$

- T_t must be constructed so that the Markov chain converge to unique stationary distribution - which is the target distribution.

Metropolis Algorithm (MA)

Given a target distribution $p(\theta \mid y)$ that can be computed up to a normalizing constant, the MA creates a sequence of random points $\theta^1, \theta^2, \ldots$ whose distributions converges to the target distribution. Each sequence can be considered a random walk whose stationary distribution is $p(\theta \mid y)$.

Metropolis Algorithm (MA)

the algorithm proceeds as follows

- draw $\theta^0 \sim p_0(\theta)$ — starting value —
- for $t = 1, 2, \ldots$
  1. draw $\theta^* \sim J_t(\theta^* \mid \theta^{t-1})$

     where the jumping distribution $J_t(\theta^* \mid \theta^{t-1})$ must be symmetric, i.e. $J_t(\theta_a \mid \theta_b) = J_t(\theta_b \mid \theta_b)$
  2. calculate the importance ratio $r = p(\theta^* \mid y)/p(\theta^{t-1} \mid y)$
  3. set $\theta^t = \theta^*$ with prob. r
- the algorithm requires the ability to calculate r and to draw θ^* from the jumping distribution $J_t(\theta^* \mid \theta)$


```
metropolis example: bivariate normal density with bivariate normal jumping kernel
metropolis_function(theta=c(1,1),mu=c(0,0),Sigma=diag(1,2)){
  thetastar_simulate.multnorm(mu,Sigma)   #jumping distribution
  Prec_solve(Sigma)
  ratio      _ exp(-1/2*(t(thetastar-mu )%*%Prec%*%(thetastar-mu) -
                        t(theta-mu)%*%Prec%*%(theta-mu)))
  u                 _ runif(1)
  if(u <= ratio) {theta_thetastar}
  test_ (u <= ratio)
  return(theta)
}

iterate_function(theta0=c(0,0),NN){
  theta_matrix(NA,2,NN)
  test_NULL
  theta[,1]_theta0
  for(n in 2:(NN-1)){
    theta[,n]_metropolis(theta=theta[,n-1],mu=c(0,0),Sigma=diag(1,2))}
  return(theta)
}
```


Why MA work ?

We want to show that

1. the simulated sequence $\theta^1, \theta^2, \ldots$ is a MC with a "unique" stationary distribution
2. the target distribution is equal to the stationary distribution

"Understanding the Metropolis-Hastings algorithm".
Chib and Greenberg, the American Statistician, 49,4,327-335

The Metropolis-Hastings algorithm (MHA)

- MHA generalizes MA because the jumping distribution needs no to be symmetric $J_t(\theta_b \mid \theta_a) \neq J_t(\theta_a \mid \theta_b)$
- to correct for the asymmetry the ratios of importance ratios is

$$r = \frac{p(\theta^* \mid y)/J_t(\theta^* \mid \theta^{t-1})}{p(\theta^{t-1} \mid y)/J_t(\theta^{t-1} \mid \theta^*)}$$

- if $J(\theta^* \mid \theta) = p(\theta^* \mid y) \forall \theta$ then $r = 1$ and θ^t are a sequence of independent draws from $p(\theta \mid y)$

Properties of a good jumping rule

- For any θ it is easy to sample from $J(\theta^* \mid \theta)$
- it is easy to calculate the ratio of importance ratios r
- each jump goes a reasonable distance in the parameters space
- the jumps are not rejected too often

The Gibbs Sampler

Alternating Conditional Sampling

- $\theta = (\theta_1, \ldots, \theta_d)$
- at each iteration t, any sub-vector θ_j of θ is sampled from the conditional distribution given all the other components of θ, i.e.

$$\begin{aligned} \theta_j^t &\sim p(\theta_j \mid \theta_{-j}^{t-1}, y) \\ \theta_{-j}^{t-1} &= (\theta_1^t, \ldots, \theta_{j-1}^t, \ldots, , \theta_{j+1}^{t-1}, \ldots, \theta_d^{t-1}) \end{aligned}$$

- each subvector is then updated conditional on the latest value of θ for the other components, which are the iteration t for the components already updated and the iteration $t-1$ values for the others

- for many problems involving standard statistical models, it is possible to sample from most or all the conditional distributions of parameters

  SAT experiments

  Diet measurements


```
Gibbs example: bivariate normal distribution:example pag:328

gibbs_function(theta1=0,theta2=0,y1=0,y2=0,rho=.81){
  theta1_rnorm(1,y1+rho*(theta2-y2),sqrt(1-rho^2))
  theta2_rnorm(1,y2+rho*(theta1-y1),sqrt(1-rho^2))
  return(theta1,theta2)
}

gibbs.iterate_function(a=0,b=0,NN){
  theta1_NULL
  theta2_NULL
  theta1[1]_a
theta2[1]_b
  for(n in 2:(NN-1)){
    theta1[n]_gibbs(theta1=theta1[n-1],theta2=theta2[n-1],y1=0,
                    y2=0,rho=.81)$theta1
    theta2[n]_gibbs(theta1=theta1[n],theta2=theta2[n-1],y1=0,
                    y2=0,rho=.81)$theta2
  }
  return(theta1,theta2)
}
```

Gibbs Sampler (GS) as special case of the MHA

- GS can be viewed as special case of MHA with the following jumping distribution

$$J_{j,t}^{Gibbs} = \begin{cases} p(\theta_j^* \mid \theta_{-j}^{t-1}, y) \text{ if } \theta_{-j}^* = \theta_{-j}^{t-1} \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases}$$

- the only possible jumps are the parameter vectors θ^* that match θ^{t-1} on all the components other than j.
- the ratio of importance ratios is

$$r = \frac{p(\theta^*|y)/J_{j,t}^{Gibbs}(\theta^*|\theta^{t-1})}{p(\theta^{t-1}|y)/J_{j,t}^{Gibbs}(\theta^{t-1}|\theta^*)}$$

$$= \frac{p(\theta^*|y)/p(\theta_j^{t-1}|\theta_{-j}^{t-1},y)}{p(\theta^{t-1}|y)/p(\theta_j^*|\theta_{-j}^{t-1},y)} = 1$$

hint: $p(\theta^* \mid y) = p(\theta^* \mid \theta_{-j}^{t-1}, y) p(\theta_{-j}^{t-1} \mid y)$

Metropolis within Gibbs

- if we cannot sample directly from all the full conditionals distributions $p(\theta_j \mid \theta_{-j}, y)$, we can approximate them by a metropolis step within the GS, i.e.

$$J_{j,t} = \begin{cases} g(\theta_j^* \mid \theta_{-j}^{t-1}) \text{ if } \theta_{-j}^* = \theta_{-j}^{t-1} \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases}$$

- the importance ratio, must be computed
- Example...

Assessing convergence

Difficulties:

1. if the iterations are not proceed long enough, the simulations maybe grossly unrepresentative of the target distribution relative to an independent sample of the same size
2. check the convergence of the chain
3. within sequence correlation: simulation inference from correlated draws is generally less precise than from the same number of independent draws

Solutions:

1. discarding early iteration (`burnin` parameter)
2. monitoring of convergence by simulating multiple sequences with starting points dispersed through the parameter space
3. using every k-th simulation draws (`skip` parameter)

Monitoring convergence of each scalar estimands

- simulate J parallel sequences of length n
- estimands $\psi_{i,j},\ i = 1, \ldots, n,\ j = 1, \ldots, J$
- let $\hat{Var}(\psi \mid y) = \frac{n-1}{n} W + \frac{1}{n} B$
- $\hat{Var}(\psi \mid y)$ overestimates the marginal posterior variance assuming the starting distribution is appropriately overdispersed
- $B = \frac{n}{J-1} \sum_{j=1}^{J} (\bar{\psi}_{.j} - \bar{\psi}_{..})^2$ — between sequence variance.
- $W = \frac{1}{J} \sum_{j=1}^{J} s_j^2,\ s_j^2 = \frac{1}{n-1} \sum_{i=1}^{n} (\psi_{ij} - \bar{\psi}_{.j})^2$ — within sequence variance
- $W \leq \hat{Var}(\psi \mid y)$ because the sequences have not had time to range over all of the target distribution

Monitoring convergence of each scalar estimands (cont's)

- the potential scale reduction is so defined

$$\sqrt{\hat{R}} = \sqrt{\hat{Var}(\psi \mid y)/W}$$

  where $\sqrt{\hat{R}} \to 1$ as $n \to \infty$. This is the factor by which the scale of the current distribution for ψ might be reduced if the simulation where continued to ∞
- calculate $\sqrt{\hat{R}}$ for all the scalar estimands and continue the simulations until $\sqrt{\hat{R}}$ is near to 1 for all of them
- `http://lib/stat.cmu/S/itsime`.
  S code that computes both the mean and the upper 95^{th} percentiles of $\sqrt{\hat{R}}$

Parametrization

- The Gibbs Sampler is most efficient when parameterized in terms of independent components
- Metropolis jumps: in a normal setting, the jumping kernel should ideally have the same covariance structure as the target distribution, which can be approximately calculated based on the normal approximation at the mode

Efficient Jumping rules

- $\theta = (\theta_1, \ldots, \theta_d)$
- $\theta \mid y \sim N_d(\mu, \Sigma)$
- $J(\theta^* \mid \theta^{t-1}) \sim N(\theta^* \mid \theta^{t-1}, c^2\Sigma)$
- among this class of jumping rules, the most efficient has scale $c \simeq 2.4/\sqrt{d}$
- for the multivariate normal distribution, the optimal jumping rule has acceptance rate .44 in one dimension and .23 in high dimensions

Adaptive simulation algorithm

- Start the parallel simulations with a fixed algorithm, such as a version of the Gibbs Sampler, or the Metropolis with a jumping rule shaped like an estimate of the target distribution (possibly covariance matrix computed at the joint or marginal mode), scaled by a factor $2.4/\sqrt{d}$
- After some number of simulations, update the Metropolis jumping rule as follows

(a) Adjust the covariance of the jumping distribution to be proportional to the posterior covariance matrix estimated from the simulations

(b) Increase or decrease the scale of the jumping distribution if the acceptance rate of the simulations is much to high or low, respectively

- the goal is to bring the jumping rule toward the approximate optimal value between .44 and ,23, depending on d.

METROPOLIS GIBBS-SAMPLER

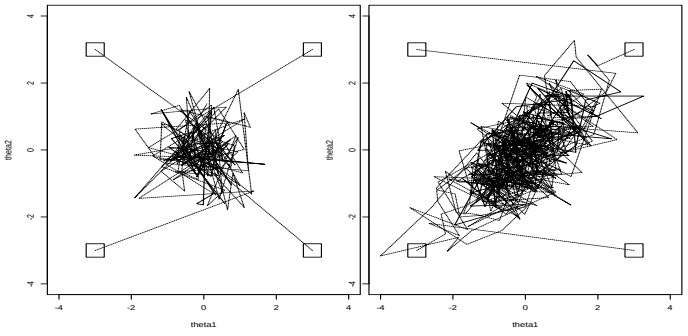

Figure 1: Four independent sequences of Metropolis and Gibbs sampler of a $N(0, I)$ and $N(0, .8I)$ with overdispersed starting points indicated by solid squares.