STAT3016/4116/7016: Introduction to Bayesian Data Analysis

RSFAS, College of Business and Economics, ANU

Non-conjugate priors and Metropolis-Hastings algorithms

Introduction

- So far the posterior distributions we have looked out have taken on a standard form, and the Gibbs sampler or the Monte Carlo method can be easily used to conduct posterior inference
- We need more general sampling methods to handle cases where non-conjugate priors are used or we cannot sample directly from the conditional distributions.
- The Metropolis-Hastings algorithm is a generic method of approximating the posterior distribution corresponding to any combination of prior distribution and sampling model.

Example - Poisson regression

- A study was conducted to assess the reproductive success of female song sparrows. Of specific interest was to study the relationship between the birds' age and the number of new offspring recorded in a summer season.
- We assume a Poisson probability model:

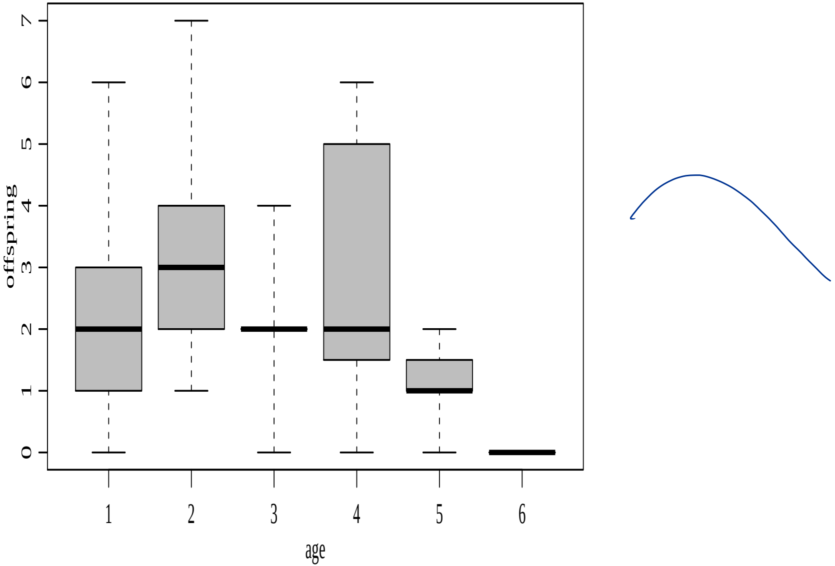
$$Y|x \sim Poisson(\theta_x)$$

usually λ

- Data on Y (number of offspring) and x (age of bird) are collected and the task is to estimate θ_x. That is, we want to estimate θ_x as a function of x. Since θ_x is a rate, we require that $\theta_x > 0$ for all x.

. linear regression is not enough.

Example - Poisson regression



Example - Poisson regression

The Poisson regression model with a log-link will meet our requirements:

ensure $\theta_x > 0$.

$$\log E[Y|x] = \log \theta_x = \beta_1 + \beta_2 x + \beta_3 x^2$$

This means that

$$Y|x \sim Poisson(\exp[\boldsymbol{\beta}^T x])$$

(The Poisson regression model is a type of *generalized linear model* (GLM), a model which relates a function of $E[Y|x]$ to a linear predictor of the form $\boldsymbol{\beta}^T x$).

$\theta_x \in [0, 1]$ log link is not appropriate.

(Another example is the logistic regression model for binary data. That is, let $Y \sim Bin(1, \theta_x)$. So $Pr(Y = 1|x) = E[Y|x] = \theta_x$ and the GLM is represented by the equation $\log \frac{\theta_x}{1-\theta_x} = \boldsymbol{\beta}^T x$).

logit is good.

Example - Poisson regression

- In a Bayesian setting, we need to assign prior distributions to the parameters $\boldsymbol{\beta}$
- What class of prior distributions would you use?
- In ordinary least squares regression, we used multivariate normal priors for $\boldsymbol{\beta}$. Can we do the same here??

The Metropolis algorithm

The posterior distribution $\boxed{p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta')p(\theta')d\theta'}}$ is often hard to calculate due to the integral in the denominator.

Ideally we want samples $\theta^{(1)}, ..., \theta^{(S)} \overset{\text{iid}}{\sim} p(\theta|y)$ and then obtain Monte-Carlo approximations to posterior quantities such as $E[g(\theta)|y] \approx \frac{1}{S}\sum_{s=1}^{S} g(\theta^{(s)})$

If we cannot sample directly from $p(\theta|y)$, then we need some alternative approach. What we want is a large collection of θ values $\{\theta^{(1)}, ..., \theta^{(S)}\}$ whose empirical distribution approximates $p(\theta|y)$. That is,

$$\frac{\#\{\theta^{(s)}\text{'s in the collection} = \theta_a\}}{\#\{\theta^{(s)}\text{'s in the collection} = \theta_b\}} \approx \frac{p(\theta_a|y)}{p(\theta_b|y)}$$

How can we construct such a collection?? Suppose we have a working collection $\{\theta^{(1)}, ..., \theta^{(s)}\}$ and we want to consider adding a new value θ^*. What would be a logical rule to guide our decision?

The Metropolis algorithm

Think about comparing $p(\theta^*|y)$ to $p(\theta^{(s)})$. Even if we cannot sample directly from $p(\theta|y)$, we can compute:

$$r = \frac{p(\theta^*|y)}{p(\theta^{(s)}|y)} = \frac{p(y|\theta^*)p(\theta^*)}{p(y)} \frac{p(y)}{p(y|\theta^{(s)})p(\theta^{(s)})} = \frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta^{(s)})p(\theta^{(s)})}$$

So we don't need to know the exact form of $p(\theta^*|y)$, we just need to derive the target posterior distribution up to a proportionality constant, that is, $p(y|\theta^*)p(\theta^*)$.

Having computed r, how should we proceed?
How do we come up with a proposed value θ^*?? Consider":

- $r < 1$
- $r > 1$

The Metropolis algorithm

1. Sample $\theta^* \sim J(\theta|\theta^{(s)})$ (where $J(\theta^*|\theta^{(s)})$ is a symmetric *proposal* distribution)
2. Compute the acceptance ratio

$$r = \frac{p(\theta^*|y)}{p(\theta^{(s)}|y)} = \frac{\overbrace{p(y|\theta^*)p(\theta^*)}^{\text{likelihood} \times \text{prior}}}{\underbrace{p(y|\theta^{(s)})p(\theta^{(s)})}_{\text{likelihood} \times \text{prior}}}$$

3. Let

$$\theta^{(s+1)} = \begin{cases} \theta^* & \text{with probability min(r,1)} \\ \theta^{(s)} & \text{with probability 1- min(r,1)} \end{cases}$$

Step 3 can be accomplished by sampling $Unif(0,1)$ random variable and setting $\theta^{(s+1)} = \theta^*$ if $u < r$ and setting $\theta^{(s+1)} = \theta^{(s)}$ otherwise.

The Metropolis algorithm - the proposal distribution

- $J(\theta^*|\theta^{(s)})$ - the proposal distribution is dependent only on the sampled value from the previous iteration. We want θ^* to be nearby the current value $\theta^{(s)}$.
- A symmetric proposal distribution means $J(\theta_b|\theta_a) = J(\theta_a|\theta_b)$
- Usually $J(\theta|\theta^{(s)}$ is very simple. Examples:
  - $J(\theta^*|\theta^{(s)}) = \text{uniform}(\theta^{(s)} - \delta, \theta^{(s)} + \delta)$.
  - $J(\theta^*|\theta^{(s)}) = \text{normal}(\theta^{(s)}, \delta^2)$

"tuning parameter"

δ^2 not too large $\Rightarrow$ not too far from $\theta^{(s)}$.
not too small $\Rightarrow$ not too close to

The Metropolis algorithm - Example

Normal distribution with known variance
Let $\theta \sim \mathrm{normal}(\mu, \tau^2)$ and $\left\{y_1, ..., y_n | \theta, \sigma^2\right\} \stackrel{\mathrm{iid}}{\sim} \mathrm{normal}(\theta, \sigma^2)$. We have learnt that $\theta | \mathbf{y} \sim \mathrm{normal}(\mu_n, \tau_n^2)$ where:

$$\mu_n = \bar{y}\left(\frac{n/\sigma^2}{n/\sigma^2 + 1/\tau^2}\right) + \mu\left(\frac{1/\tau^2}{n/\sigma^2 + 1/\tau^2}\right)$$

$$\tau_n^2 = \frac{1}{n/\sigma^2 + 1/\tau^2}$$

Simulation: $\sigma^2 = 1$; $\tau^2 = 10$; $\mu = 5$; $n = 5$ and $\mathbf{y} = (9.37, 10.18, 9.16, 11.60, 10.33)$. For these data $\mu_n = 10.03$ and $\tau_n^2 = 0.20$.

The Metropolis algorithm - Example

Normal distribution with known variance

Suppose we weren't able to derive and sample directly from the exact posterior distribution. So we use the Metropolis algorithm to approximate it. The acceptance ratio is

normal den

$$\boxed{r = \frac{\prod_{i=1}^{n} p(y_i|\theta^*, \sigma^2)}{\prod_{i=1}^{n} p(y_i|\theta^{(s)}, \sigma^2)} \times \frac{p(\theta^*, \mu, \tau^2)}{p(\theta^{(s)}, \mu, \tau^2)}}$$

r is easy to calculate.

The acceptance ratio r can be unstable and it is preferable to work with log r. On the log scale, the proposal is accepted if

$$\boxed{\log u < \log r \ (\text{for } u \sim Unif(0,1))}$$

unstable:

keep accepting

The Metropolis algorithm - Example

Normal distribution with known variance

```
n<-5
y<-c(9.37, 10.18, 9.16, 11.60, 10.33)

mu.n<-( mean(y)*n/s2 + mu/t2 )/( n/s2+1/t2)
t2.n<-1/(n/s2+1/t2)

#####
s2<-1 ; t2<-10 ; mu<-5

theta<-0 ; delta<-2 ; S<-10000 ; THETA<-NULL
```

The Metropolis algorithm - Example

Normal distribution with known variance

```
for(s in 1:S)
{

  theta.star<-rnorm(1,theta,sqrt(delta))

  log.r<-( sum(dnorm(y,theta.star,sqrt(s2),log=TRUE)) +
               dnorm(theta.star,mu,sqrt(t2),log=TRUE) ) -
         ( sum(dnorm(y,theta,sqrt(s2),log=TRUE)) +
               dnorm(theta,mu,sqrt(t2),log=TRUE) )

  if(log(runif(1))<log.r) { theta<-theta.star }

  THETA<-c(THETA,theta)

}
```

b/c it's on log scale.

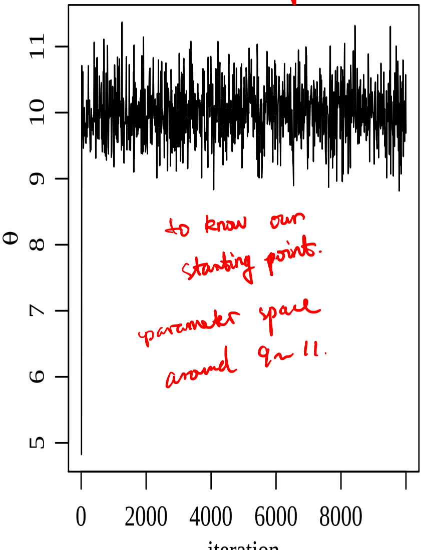
if log(runif(1)) ≥ log.r

we keep the old theta.

The Metropolis algorithm - Example

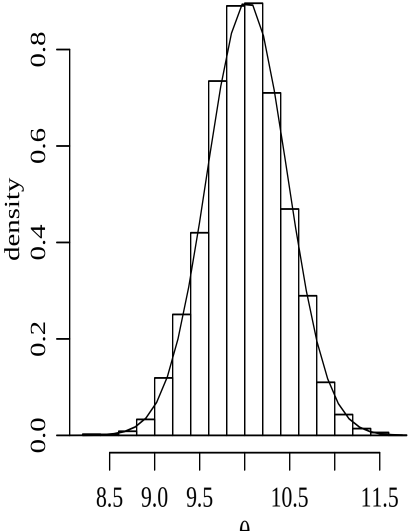
Normal distribution with known variance

trace plot.



to know our starting point.

parameter space around 9~11.



The Metropolis algorithm - output

The dependent sequence $\left\{\theta^{(1)}, \theta^{(2)}.....\right\}$ forms a Markov Chain.

The marginal sampling distribution of $\theta^{(s)}$ is approximates $p(\theta|y)$ for large S. Also, for any given numerical value θ_a of θ:

$$\lim_{x\to\infty} \frac{\#\left\{\theta\text{'s in the sequence} < \theta_a\right\}}{S} = p(\theta < \theta_a|y)$$

So we can approximate posterior quantities using the empirical distribution of $\left\{\theta^{(1)}, \theta^{(2)}.....\theta^{(S)}\right\}$. As $S \to \infty$, the approximation will be exact. How large should S be??

The Metropolis algorithm - output

1. Run algorithm until some iteration B for which it looks like the Markov chain has achieved stationarity
2. Run the algorithm S more times, generating $\{\theta^{(B+1)}, \theta^{(B+2)} \ldots\ldots \theta^{(B+S)}\}$
3. Discard $\{\theta^{(1)}, \theta^{(2)} \ldots\ldots \theta^{(B)}\}$ and use the empirical distribution of $\{\theta^{(B+1)}, \theta^{(B+2)} \ldots\ldots \theta^{(B+S)}\}$ to approximate $p(\theta|y)$.

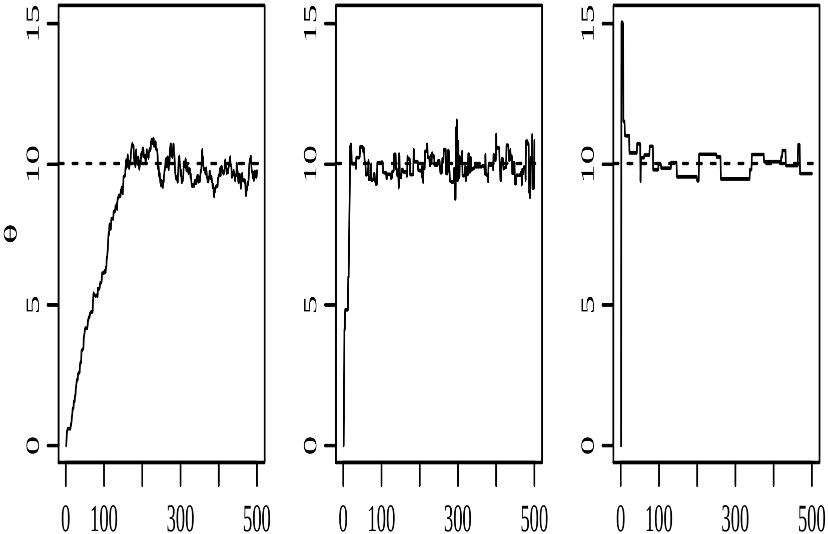
Burn-in period - iterations 1 to B (inclusive). In the burn-in period, the Markov chain moves from its initial value to a region of the parameter space that has high posterior probability. If you know where this region is, it is a good a idea to start your Markov Chain there.

The Metropolis algorithm - output

- What happens to the Metropolis algorithm output when $\left\{\theta^{(1)}, \theta^{(2)} .....\theta^{(s)}\right\}$ are highly correlated?
- How can we decrease the correlation in the Markov Chain??

The Metropolis algorithm - output

Three values for δ are assumed (1/32,2,64). Which figure corresponds to which value of δ?



The Metropolis algorithm - output

Choosing a value of δ (tuning parameter)

- The proposal variance should be large enough so that the Markov chain can quickly move around the parameter space
- The proposal variance should not be so large that proposals end up getting rejected most of the time.
- Select δ for an acceptance rate of between 20% and 50%.
- δ can be found by trialling several different values for short runs or adaptively updating δ.

$$20\% < \delta < 50\%$$

The Metropolis algorithm - Poisson regression example

Recall the model $Y_i \sim \mathrm{Poisson}(\exp(\boldsymbol{\beta}^T x_i))$. That is

$$\log E[Y_i|x_i] = \beta_1 + \beta_2 age_i + \beta_3 age_i^2$$

The parameters of interest are the $\boldsymbol{\beta}$'s. We want to obtain posterior estimates of the $\boldsymbol{\beta}$'s and we will run a Metropolis algorithm. What is the acceptance ratio??

20% ~ 50%.

$$r = ??$$

What about the proposal distribution $J(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(s)})$??

multivariate distribution.

(like multivariate normal).

The Metropolis algorithm - Poisson regression example

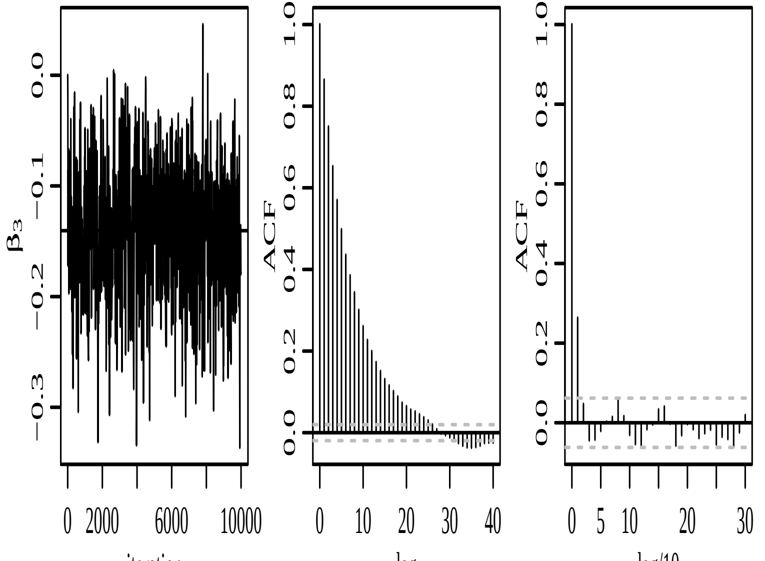
```
yX.sparrow<-data.frame(yX.sparrow)
attach(yX.sparrow)
y<-fledged ;
n<-length(y)
X<-cbind(rep(1,n),age,age^2)
yX<-cbind(y,X)
p<-dim(X)[2]

pmn.beta<-rep(0,p)  #prior expectation
psd.beta<-rep(10,p)  #prior variance

var.prop<- var(log(y+1/2))*solve( t(X)%*%X )
beta<-rep(0,p)
S<-10000
BETA<-matrix(0,nrow=S,ncol=p)
ac<-0
```

The Metropolis algorithm - Poisson regression example

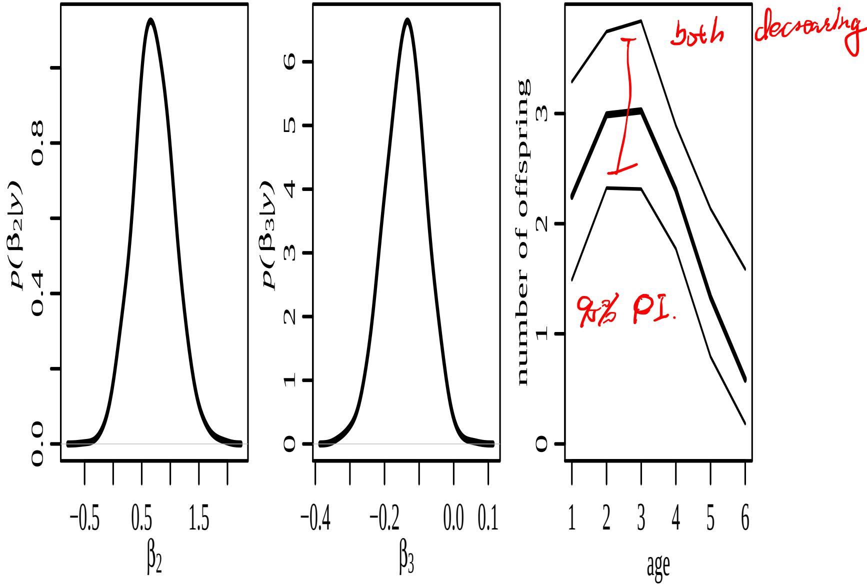
```
for(s in 1:S) {

#propose a new beta
beta.p<- t(rmvnorm(1, beta, var.prop ))
lhr<- sum(dpois(y,exp(X%*%beta.p),log=T)) -
      sum(dpois(y,exp(X%*%beta),log=T)) +
      sum(dnorm(beta.p,pmn.beta,psd.beta,log=T)) -
      sum(dnorm(beta,pmn.beta,psd.beta,log=T))

if( log(runif(1))< lhr ) { beta<-beta.p ; ac<-ac+1 }
BETA[s,]<-beta
                    }
cat(ac/S,"\n")
0.4293
> library(coda)
> apply(BETA,2,effectiveSize)
theta.star[1] 818.4049 778.4707 726.3633
```

The Metropolis algorithm - Poisson regression example

The right panel is the acf for the ‘thinned’ sequence (plots every 10th value of β_3)



The Metropolis algorithm - Poisson regression example



Metropolis, Metropolis-Hastings and Gibbs

Suppose our target probability distribution is $p_0(u, v)$, a bivariate distribution of two random variables U and V. (For example, $U = \theta$ and $V = \sigma^2$)

- How would we apply the Gibbs sampler approach to approximate the target distribution $p_0(u, v) = p(\theta, \sigma^2|y)$?
- How would we apply the Metropolis algorithm to approximate the target distribution $p_0(u, v) = p(\theta, \sigma^2|y)$?

The Metropolis-Hastings algorithm generalises both the Gibbs sampler and Metropolis algorithm by allowing arbitrary proposal distributions (that is, not necessarily symmetric or with guaranteed acceptance). The proposal distributions can be symmetric around the current values, full conditional distributions, or something else entirely.

The Metropolis-Hastings algorithm

(both)

conditional on past value of u & v.

1. update U:
   (a) sample $u^* \sim J_u(u|u^{(s)}, \nu^{(s)})$
   (b) compute the acceptance ratio

post. likelihood

$$r = \frac{p_0(u^*, \nu^{(s)})}{p_0(u^{(s)}, \nu^{(s)})} \times \frac{J_u(u^{(s)}|u^*, \nu^{(s)})}{J_u(u^*|u^{(s)}, \nu^{(s)})}$$

   (c) set $u^{(s+1)}$ to u^* with probability min(1,r).
2. update V:
   (a) sample $\nu^* \sim J_\nu(\nu|u^{(s+1)}, \nu^{(s)})$
   (b) compute the acceptance ratio

$$r = \frac{p_0(u^{(s+1)}, \nu^*)}{p_0(u^{(s+1)}, \nu^{(s)})} \times \frac{J_\nu(\nu^{(s)}|u^{(s+1)}, \nu^*)}{J_\nu(\nu^*|u^{(s+1)}, \nu^{(s)})}$$

   (c) set $\nu^{(s+1)}$ to ν^* with probability min(1,r).

The Metropolis-Hastings algorithm

- Note that J_u and J_ν need not be symmetric
- Notice that the acceptance ratio contains an extra factor: $\frac{J_u(u^{(s)}|u^*,\nu^{(s)})}{J_u(u^*|u^{(s)},\nu^{(s)})}$. Provide an interpretation for this extra factor.
- For the Metropolis algorithm: $J(u_a|u_b,\nu) = J(u_b|u_a,\nu)$ for all possible values of u_a, u_b and ν, so the correction factor is 1.
- Exercise: Show how the Gibbs sampler is a special case of the Metropolis Hastings algorithm. (Hint: What is the jumping distribution for the Gibbs sampler?)

The Metropolis-Hastings algorithm - why does it work?

Let our target distribution be $p_0(x)$. Let $x^{(s)}$ be a current value of X.

1. Generate x^* from $J_s(x^*|x^{(s)})$;
2. Compute the acceptance ratio

$$r = \frac{p_0(x^*)}{p_0(x^{(s)})} \times \frac{J_s(x^{(s)}|x^*)}{J_s(x^*|x^{(s)})}$$

3. Sample $u \sim \mathrm{uniform}(0, 1)$. If $u < r$ set $x^{(s+1)} = x^*$, else set $x^{(s+1)} = x^{(s)}$

The Metropolis-Hastings algorithm - why does it work?

Requirements of $J_s(x^*|x^{(s)})$:

- Does not depend on values in the sequence previous to $x^{(s)}$.
- Choose J_s so that the Markov chain is able to converge to the target distribution $p_0(x)$. This requires the Markov chain to be *irreducible* and *aperiodic* and *recurrent*.

Theorem (Ergodic Theorem): If $\{x^{(1)}, x^{(2)}, ..., \}$ is an irreducible, aperiodic and recurrent Markov chain, then there is a unique probability disitribution π such that as $s \to \infty$,

- $Pr(x^{(s)} \in A) \to \pi(A)$ for any set A.
- $\frac{1}{S} \sum g(x^{(s)}) \to \int g(x)\pi(x)dx$

The distribution π is called the *stationary* distribution. This means

If $x^{(s)} \sim \pi$

and $x^{(s+1)}$ is generated from the Markov chain starting at $x^{(s)}$, then $Pr(x^{(s+1)} \in A) = \pi(A)$

The Metropolis-Hastings algorithm - why does it work?

Prove that $\pi(x) = p_0(x)$:

We need to show that $p_0(x)$ is the unique stationary distribution.

Suppose $x^{(s)}$ is sampled from the target distribution $p_0(x)$ and then $x^{(s+1)}$ is generated from $x^{(s)}$ using the MH algorithm. We need to show that $Pr(x^{(s+1)} = x) = p_0(x)$ (as required by the Ergodic theorem above).

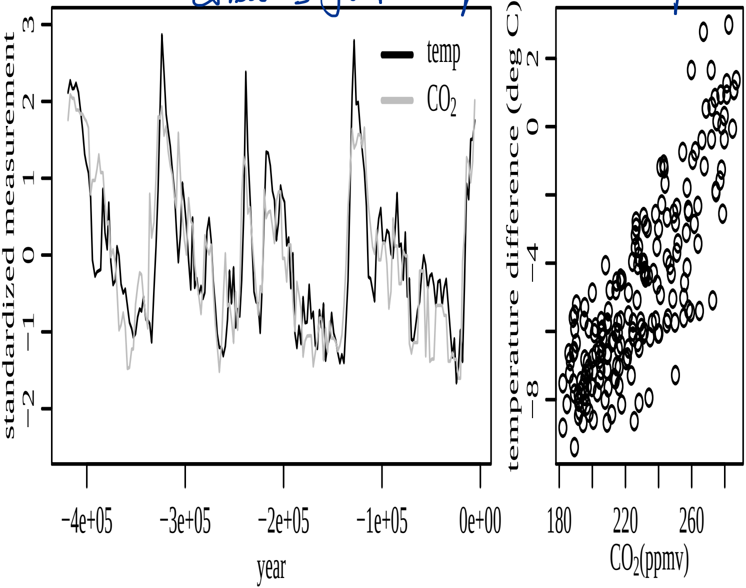
Let x_a and x_b be any two values of X such that
$p_0(x_a)J_s(x_b|x_a) > p_0(x_b)J_s(x_a|x_b)$ or $<$ or $=$

- Write down the probability that $x^{(s)} = x_a$ and $x^{(s+1)} = x_b$ under the Metropolis-Hastings algorithm.
- Write down the probability that $x^{(s)} = x_b$ and $x^{(s+1)} = x_a$ under the Metropolis-Hastings algorithm.
- Use your result to derive the marginal probability $Pr(x^{(s+1)} = x)$

Combining the Metropolis and Gibbs algorithm

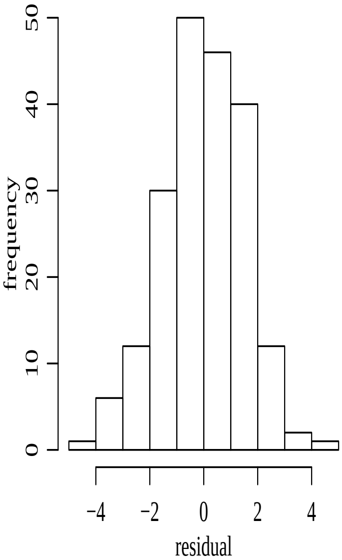
Example: Historical CO_2 and temperature data

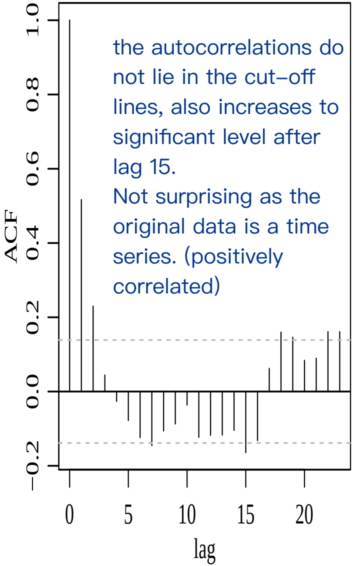
Gibbs is just a special case of M-H.



Combining the Metropolis and Gibbs algorithm

Example: Historical CO_2 and temperature data





the autocorrelations do not lie in the cut–off lines, also increases to significant level after lag 15.
Not surprising as the original data is a time series. (positively correlated)

Combining the Metropolis and Gibbs algorithm

Example: Historical CO_2 and temperature data: A regression model with correlated errors.
The ordinary regression model is

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} \sim \text{multivariate normal}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathrm{I})$$

The diagnostic plots of the ice core data suggest that the errors are temporally correlated. We want to replace the covariance matrix $\sigma^2\mathrm{I}$ with a matrix Σ that can represent positive correlation between sequential observations.

Combining the Metropolis and Gibbs algorithm

Example: Historical CO_2 and temperature data

Consider a first-order autoregressive structure:

$$\Sigma = \sigma^2 \mathbf{C}_p = \sigma^2 \begin{pmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{n-1} \\ \rho & 1 & \rho & \cdots & \rho^{n-2} \\ \vdots & \vdots & \ddots & \vdots & \\ \rho^{n-1} & \rho^{n-2} & \rho & \cdots & 1 \end{pmatrix}$$

Combining the Metropolis and Gibbs algorithm

Example: Historical CO_2 and temperature data

Using multivariate normal and inverse-gamma prior distributions for β and σ^2 (see slides on linear regression) we can show that

Cp works in general, but for this question we have a specific Cp as written in the previous

$$\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}, \sigma^2, \rho \sim \text{multivariate normal}(\boldsymbol{\beta_n}, \Sigma_n) \text{ where}$$

if Cp is an identical matrix, then posterior reduces to posterior of beta in linear regression model

$$\Sigma_n = (X^T C_p^{-\mathbf{1}} X/\sigma^2 + \Sigma_0^{-1})^{-1}$$

$$\boldsymbol{\beta_n} = \Sigma_n (X^T C_p^{-\mathbf{1}} y/\sigma^2 + \Sigma_0^{-1} \boldsymbol{\beta_0}) \text{ and}$$

$$\sigma^2 | X, y, \boldsymbol{\beta}, \rho \sim \text{inverse-gamma}([\nu_0 + n]/2, [\nu_0 \sigma_0^2 + SSR_\rho]/2) \text{ where}$$

$$SSR_\rho = ((y - X\boldsymbol{\beta})^T C_{\boldsymbol{\rho}}^{-\mathbf{1}} (y - X\boldsymbol{\beta}))$$

only difference is that SSRp is now a weighted sum of errors

Combining the Metropolis and Gibbs algorithm

Example: Historical CO_2 and temperature data
If we knew ρ then we could approximate $p(\boldsymbol{\beta}, \sigma^2 | \boldsymbol{X}, y, \rho)$ using the Gibbs sampler and iteratively sample from the full conditional distributions given on the previous slide.

For most prior distributions of ρ the full conditional distrbution for ρ will be non-standard. The Gibbs sampler is not applicable here to estimate ρ, let's use the Metropolis algorithm to obtain posterior draws of ρ.

The combined (MH) algorithm is as follows:
Given $\left\{\boldsymbol{\beta}^{(s)}, \sigma^{2(s)}, \rho^{(s)}\right\}$, generate a new set of values

1. Update $\boldsymbol{\beta}$: Sample $\boldsymbol{\beta}^{(s+1)} \sim \text{multivariate normal}(\boldsymbol{\beta}_n, \Sigma_n)$ where $\boldsymbol{\beta}_n$ and Σ_n depend on $\sigma^{2(s)}$ and $\rho^{(s)}$
2. Update σ^2: Sample $\sigma^{2(s+1)} \sim \text{inverse-gamma}([\nu_0 + n]/2, [\nu_0\sigma_0^2 + SSR_\rho]/2)$ where SSR_p depends on $\boldsymbol{\beta}^{(s+1)}$ and $\rho^{(s)}$

Combining the Metropolis and Gibbs algorithm

not strictly symmetric, but we assume it is.
for most rhos we generated from the algorithm,
they are symmetric.

3. Update ρ:
   a. Propose $\rho^* \sim \mathrm{uniform}(\rho^{(s)} - \delta, \rho^{(s)} + \delta)$. If $\rho^* < 0$ then reassign it to be $|\rho^*|$. If $\rho^* > 1$ reassign it to be $2 - \rho^*$. (nb: this is a reflecting random walk)
   b. Compute the acceptance ratio

$$r = \frac{p(y|X, \boldsymbol{\beta}^{(s+1)}, \sigma^{2(s+1)}, \rho^*)p(\rho^*)}{p(y|X, \boldsymbol{\beta}^{(s+1)}, \sigma^{2(s+1)}, \rho^{(s)})p(\rho^{(s)})}$$

   and sample $u \sim \mathrm{uniform}(0, 1)$. If $u < r$, set $\rho^{(s+1)} = \rho^*$, otherwise set $\rho^{(s+1)} = \rho^{(s)}$

Combining the Metropolis and Gibbs algorithm

Analysis of the ice core data

Assume diffuse priors $\beta_0 = \mathbf{0}$, Σ_0=diag(1000), $\nu_0 = 1$ and $\sigma_0^2 = 1$.
Our prior for ρ will be the uniform distribution on (0,1).

```
nu0<-1 ; s20<-1 ; T0<-diag(1/1000,nrow=2)
###
set.seed(1)
S<-25000 ; odens<-S/1000
OUT<-NULL ; ac<-0 ; par(mfrow=c(1,2))
for(s in 1:S)
{

  Cor<-phi^DY  ; iCor<-solve(Cor)
  V.beta<- solve( t(X)%*%iCor%*%X/s2 + T0)
  E.beta<- V.beta%*%( t(X)%*%iCor%*%y/s2  )
  beta<-t(rmvnorm(1,E.beta,V.beta)  )

  s2<-1/rgamma(1,(nu0+n)/2,(nu0*s20+t(y-X%*%beta)%*%
         iCor%*%(y-X%*%beta)) /2 )
```

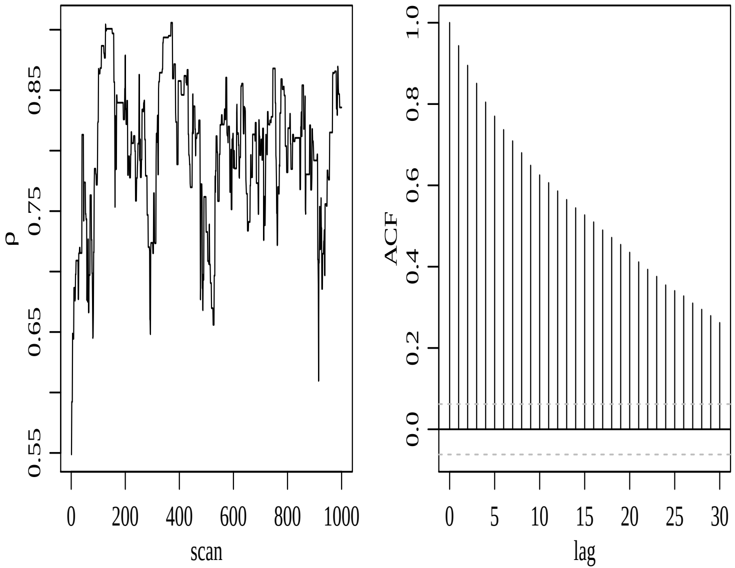
Combining the Metropolis and Gibbs algorithm

Analysis of the ice core data

```
  phi.p<-abs(runif(1,phi-.1,phi+.1))
  phi.p<- min( phi.p, 2-phi.p)
  lr<- -.5*( determinant(phi.p^DY,log=TRUE)$mod -
             determinant(phi^DY,log=TRUE)$mod  +
   tr( (y-X%*%beta)%*%t(y-X%*%beta)%*%(solve(phi.p^DY) -solve(ph

  if( log(runif(1)) < lr ) { phi<-phi.p ; ac<-ac+1 }

  if(s%%odens==0)
    {
      cat(s,ac/s,beta,s2,phi,"\n") ;
      OUT<-rbind(OUT,c(beta,s2,phi))

    }
}
```

Combining the Metropolis and Gibbs algorithm

Analysis of the ice core data



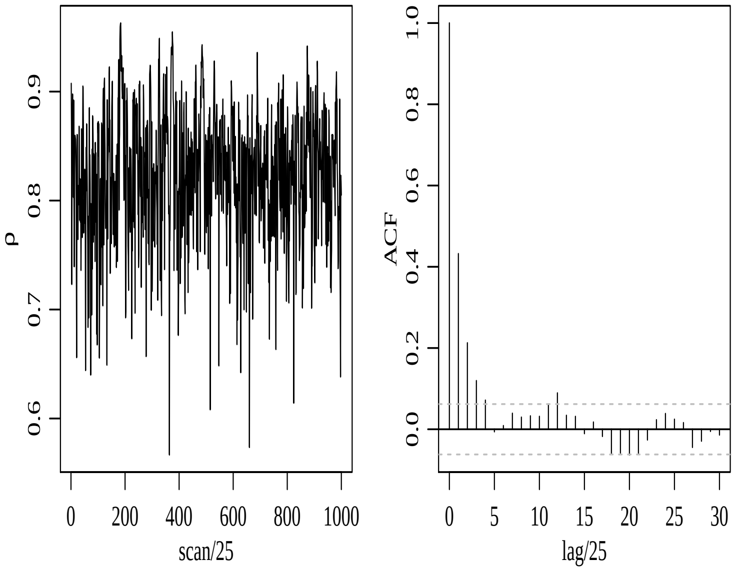
Combining the Metropolis and Gibbs algorithm

Analysis of the ice core data

```
> apply(OUT.25000,2,effectiveSize )
[1] 52.04899 50.77088 20.17433 23.40762

> apply(OUT.1000,2,effectiveSize )

   695.1691 682.4899 191.4987 396.1966
```

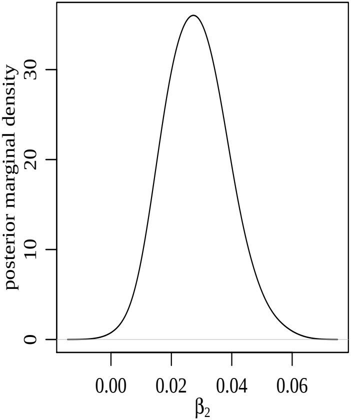
Combining the Metropolis and Gibbs algorithm

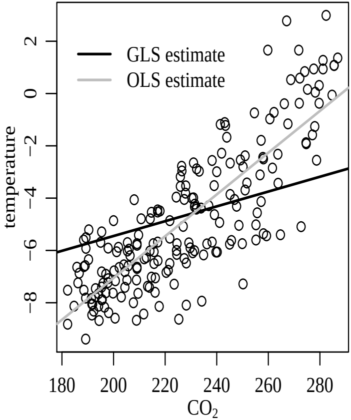
Analysis of the ice core data



Combining the Metropolis and Gibbs algorithm

Analysis of the ice core data






```
quantile(OUT.1000[,2],probs=c(.025,.975) )
      2.5%      97.5%
0.01133394 0.04838844
> mean(OUT.1000[,2])
[1] 0.02832241
```