Estatística Computacional

Inferência Bayesiana - MCMC - Parte 3

Prof. Paulo Cerqueira Jr
Faculdade de Estatística - FAEST
Programa de Pós-graduação em Matemática e Estatística - PPGME

https://github.com/paulocerqueirajr

Introdução

Introdução

- Deseja-se construir um algoritmo de simulação de uma Cadeia de Markov (CM) cuja distribuição limite (estacionária) é alguma distribuição de interesse h, que é:

  1. multidimensional;

  2. e geralmente conhecemos somente o núcleo desta distribuição.

- O interesse principal é obter amostras das distribuições marginais envolvidas em h.
- A ideia é que se deixarmos a CM evoluir, obtemos a partir de um certo tempo T_0 uma amostra aproximada da distribuição conjunta h e também das distribuições marginais de interesse.
- Principais algoritmos: Amostrador de Gibbs e Metropolis-Hastings.

Amostrador de Gibbs

Amostrador de Gibbs

- A construção do algoritmo depende do conhecimento das distribuições condicionais completas.
- A partir da simulação destas distribuições de forma iterativa, podemos obter amostras aproximadas das distribuições marginais.
- Considere o vetor aleatório $\theta = (\theta_1, \theta_2, \ldots, \theta_d)$ com função densidade de probabilidade (ou f.p., no caso discreto) conjunta h e seja

$$h(\theta_j \mid \theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_d),$$

a densidade da $j-$ ésima componente condicional às demais componentes do vetor.

- Essas distribuições são denominadas .

Amostrador de Gibbs

- Suponha que é possível obter um algoritmo de simulação eficiente para simular amostras dessas distribuições.
- Usualmente, em Inferência Bayesiana, é possível obter o núcleo da distribuição conjunta `a posteriori`, mas é difícil obter as distribuições `a posteriori` marginais.
- No caso das distribuições `a posteriori` condicionais terem formas conhecidas (de fácil simulação), o pode ser facilmente implementado.

O algorítmo

1. Fixar um conjunto de valores iniciais $\theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_d^{(0)})$.

2. Obter um novo valor $\theta^{(j)}(\theta_1^{(j)}, \theta_2^{(j)}, \ldots, \theta_d^{(j)})$ a partir de um valor $\theta^{(j-1)}$ usando as etapas:

   a. Gerar $\theta_1^{(j)} \sim h(\theta_1 \mid \theta_2^{(j-1)}, \ldots, \theta_d^{(j-1)})$;

   b. Gerar $\theta_2^{(j)} \sim h(\theta_2 \mid \theta_1^{(j)}, \theta_3^{(j-1)}, \ldots, \theta_d^{(j-1)})$;

   $$\vdots$$

   c. Gerar $\theta_d^{(j)} \sim h(\theta_d \mid \theta_1^{(j)}, \theta_2^{(j)}, \ldots, \theta_{d-1}^{(j)})$.

- Mudar o contador de j para $j + 1$ e voltar a etapa 2 até obter a convergência.
- Quando a convergência é atingida, o valor obtido $\theta^{(j)} = (\theta_1^{(j)}, \theta_2^{(j)}, \ldots, \theta_d^{(j)})$ é uma amostra da distribuição h.
- Quando o número de iterações cresce, a cadeia se aproxima do equilíbrio, então assume-se que a convergência ocorreu aproximadamente.

Exemplo

No caso do modelo normal quando tem-se $X = (X_1, X_2, \ldots, X_n)$ uma amostra aleatória de $X \sim N(\mu, 1/\phi)$, temos a seguinte distribuição `a posteriori`

$$h(\mu, \phi \mid X) \propto \phi^{n/2+a-1} \exp\{-\phi B\} \times (2\pi V)^{-1/2} \exp\left\{-\frac{1}{2V}(\mu - M)^2\right\}$$

Neste caso, as condicional completa para μ:

$$h(\mu, \mid \phi, X) \propto (2\pi V)^{-1/2} \exp\left\{-\frac{1}{2V}(\mu - M)^2\right\}$$

e para ϕ:

$$h(\phi \mid \mu, X) \propto \phi^{n/2+a-1} \exp\{-\phi B\}$$

JAGS (Just Another Gibbs Sampler)

JAGS (Just Another Gibbs Sampler)

- JAGS (Plummer, 2011) é Just Another Gibbs Sampler que foi escrito principalmente por Martyn Plummer para fornecer um motor BUGS para Unix.

Alguns pacotes em R:

- `R2jags` (Su e Yajima, 2012) é um pacote R que permite ajustar modelos JAGS de dentro de R. Quase todos os exemplos em Análise de dados de Gelman e Hill usando regressão e multinível/hierárquico Models (2007) pode ser trabalhado de forma equivalente em JAGS, usando R2jags.
- `rjags` (Plummer, 2013) é outro pacote R que permite ajustar modelos JAGS de dentro do R. R2jags depende disso. A Análise Bayesiana de Simon Jackman para as Ciências Sociais (2009) fornece muitos exemplos usando rjags, assim como Doing Bayesian Data Analysis (2011), de John Kruschke.
- `runjags` (Denwood, N.d.) permite algumas funcionalidades adicionais, incluindo computação paralela.

JAGS (Just Another Gibbs Sampler)

- O JAGS é um compilador que permite realizar análises bayesianas que precisa ser instalado de forma independente do R.
- Os passos de instalação:

1. Instalar a versão mais recente do R-Mac ou R-Win
2. Instalar o JAGS, versão 3.4.0 do repositório de Martyn Plummer: JAGS

Exemplo:

```r
require(runjags)
testjags()
```

```
You are using R version 4.4.0 (2024-04-24 ucrt) on a windows machine,
with the RTerm GUI
JAGS version 4.3.1 found successfully using the command 'C:/Program
Files/JAGS/JAGS-4.3.1/x64/bin/jags-terminal.exe'
The rjags package is installed
```

Exemplo: Modelagem Gaussiana

- Gerando os dados:

```
set.seed(432104)
n <- 1000
x <- rnorm(n, 0, 5)
```

- Definição do modelo:

```
model.Gauss <-"

model {

    # Verossimilhança:
    for (i in 1:n){
    x[i] ~ dnorm(mu, phi)
    }

  # Priori:
  mu ~ dnorm(0,.0001)
  phi <- pow(sigma, -2)
  sigma ~ dunif(0,100)
}
"
```

Modelagem Gaussiana

> Configurações do `Amostrador de Gibbs`:

```r
dados <- list(x=x, n=n)

inits.gen <- list(mu=1, sigma=100)

param<- c("mu", "sigma", "phi")

runjags.options(method = "rjags")

jagsfit <- run.jags(model = model.Gauss,
                    monitor = param,
                    data = dados, inits = inits.gen,
                    adapt = 1000, n.chains = 1, thin = 1,
                    burnin = 100, sample = 1000)
```

Modelagem Gaussiana

> Configurações do `Amostrador de Gibbs`:

```r
dados <- list(x=x, n=n)

inits.gen <- list(mu=1, sigma=100)

param<- c("mu", "sigma", "phi")

runjags.options(method = "rjags")

jagsfit <- run.jags(model = model.Gauss,
                    monitor = param,
                    data = dados, inits = inits.gen,
                    adapt = 1000, n.chains = 1, thin = 1,
                    burnin = 100, sample = 1000, silent.jags = TRUE)
```

```
Finished running the simulation
```

Modelagem Gaussiana

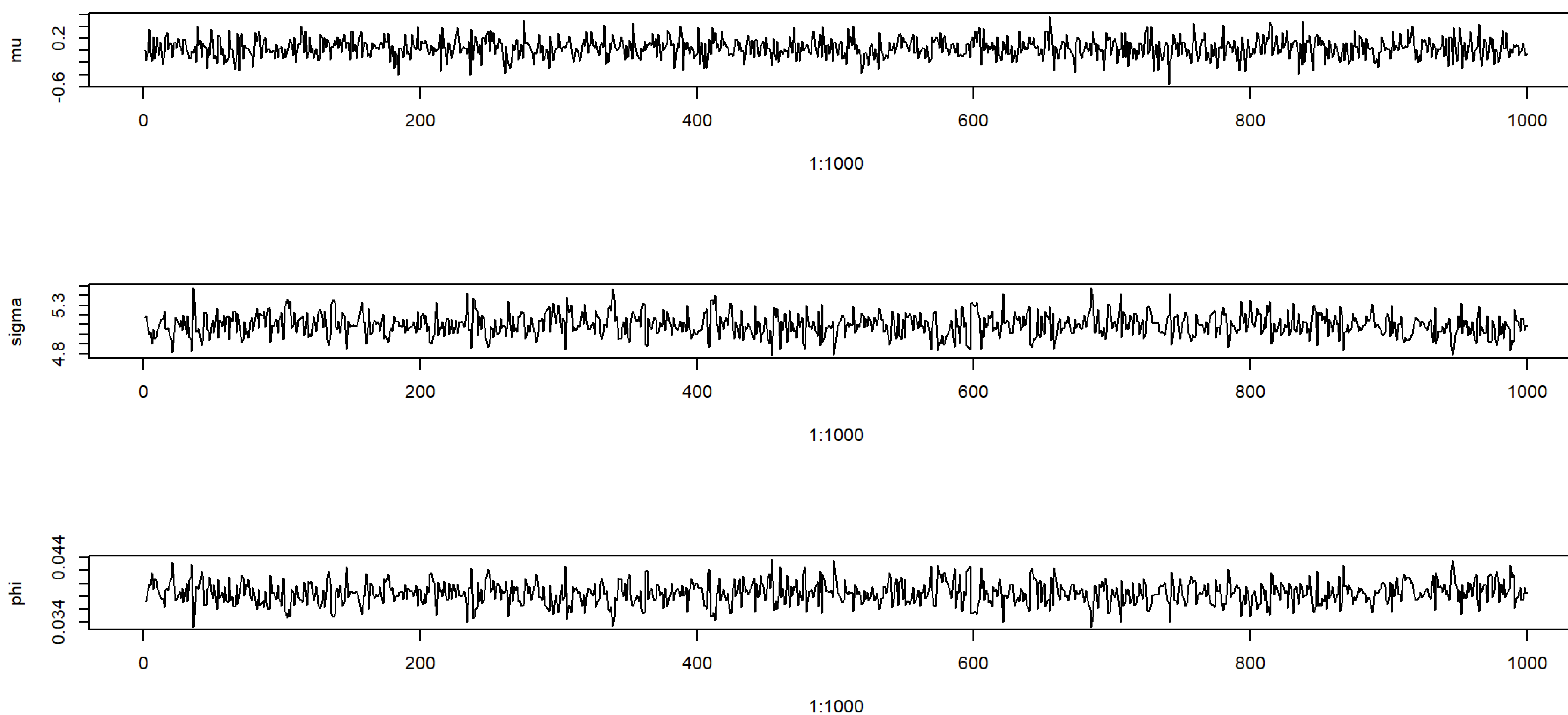
> Resumo descritivo da amostra `a posteriori`:

```r
jagsfit$summaries
```

```
         Lower95     Median    Upper95       Mean          SD Mode
mu    -0.25418093 0.04472057 0.39629698 0.04495132 0.162112915   NA
sigma  4.88180922 5.09456545 5.33132539 5.09844033 0.116410252   NA
phi    0.03518274 0.03852882 0.04196028 0.03853026 0.001755073   NA
             MCerr MC%ofSD SSeff        AC.10 psrf
mu    5.126460e-03     3.2  1000  0.006953886   NA
sigma 4.799784e-03     4.1   588 -0.009542474   NA
phi   7.223208e-05     4.1   590 -0.009119493   NA
```

Modelagem Gaussiana

```r
par(mfrow=c(3,1))
plot(1:1000, unlist(jagsfit$mcmc[,"mu"]), type="l", ylab="mu")
plot(1:1000, unlist(jagsfit$mcmc[,"sigma"]), type="l", ylab="sigma")
plot(1:1000, unlist(jagsfit$mcmc[,"phi"]), type="l", ylab="phi")
```



Suporte computacional: pacote CODA (no R)

- Faz gráficos dos valores simulados (trace plots), de auto correlações, de medidas descritivas ergódigas, de estatística de ajuste, auto-correlações.
- Calcula estimativas Bayesianas pontuais e intervalares.
- Uma única amostra de cada parâmetro: matriz em que as linhas são os valores simulados e as colunas são os parâmetros. Transformas num objeto `mcmc`.
- Várias amostras de cada parâmetro: concatenar verticalmente objetos `mcmc` num objeto `mcmc.list`.

Algorítmo de Metropolis-Hastings

- O que fazer quando não conhecemos a forma das distribuições condicionais completas (no sentido de que não a reconhecemos como uma distribuição conhecida)?
- Tem-se que se utilizar algum algoritmo auxiliar para simular dessa densidade:

  1. Metropolis-Hastings (Metropolis-Hastings dentro do amostrador de Gibbs);
  2. rejeiçao adaptativa;
  3. amostragem por importância;
  4. amostragem por corte “slice sampling”).

- Para as densidades completas que não são conhecidas (e/ou difícieis de simular).

Algorítmo de Metropolis-Hastings

- O algoritmo depende de um núcleo de transição proposto Q e também de uma probabilidade de aceitação $\alpha(\theta, \phi)$ para o valor simulado ϕ, dado que a cadeia está em θ.
- Considere o vetor aleatório $\theta = (\theta_1, \theta_2, \ldots, \theta_d)$ com função densidade de probabilidade (ou f.p., no caso discreto) conjunta h e um núcleo de transição Q associado a uma CM da qual sabemos simular.

Algorítmo de Metropolis-Hastings

a. Fixar um conjunto de valores iniciais $\theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_d^{(0)})$;

b. Obter um novo valor ϕ simulado de $Q(\theta^{(j-1)}, \cdot)$;

c. Calcular a probabilidade de aceitação

$$\alpha(\theta, \phi) = \min\left\{1, \frac{h(\phi)Q(\phi, \theta)}{h(\theta)Q(\theta, \phi)}\right\};$$

d. Mover a cadeia para $\theta^{(j)} = \phi$ se $u < \alpha$, em que $u \sim U(0, 1)$. Caso contrário, fazer $\theta^{(j)} = \theta^{(j-1)}$;

e. Mudar o contador de j para $j + 1$ e voltar a etapa (b) até obter a convergência.

Escolha da distribuição proposta Q:

i. Proposta simétrica: (versão Metropolis).

$$Q(x, y) = Q(y, x), \forall x, y \in S.$$

- Nesse caso,

$$\alpha(\theta, \phi) = \min \left\{1, \frac{h(\phi)}{h(\theta)}\right\};$$

- Um exemplo é o uso de um Passeio aleatório: $\theta^{(j)} = \theta^{(j-1)} + \epsilon$;
- Se $\theta \in \mathrm{R}^d$ é usual considerar $\epsilon \sim N_d(0, cV)$.
- A matriz V pode ser definida utilizando-se alguma aproximação da matriz de covariâncias a posteriori.
- O valor c é denominado *tuning constant* (constante de afinação) e pode ser monitorada.

Escolha da distribuição proposta Q:

ii. Proposta Independente:

- A distribuição proposta independe da posição atual da CM, isto é,

$$Q(\theta, \phi) = f(\phi),$$

resultando na seguinte probabilidade de aceitação

$$\alpha(\theta, \phi) = \min \{1, w(\phi)/w(\phi)\},$$

onde $w(x) = h(x)/f(x)$ representa o peso associado ao valor x.

Obs: Se usarmos como proposta a distribuição `a priori`, os pesos w serão dados pela razão de verossimilhanças.

Tamanho efetivo da amostra e autocorrelações

- A autocorrelação de comprimento k(lag k) da CM

$$\rho = \frac{Cov\left(t^{(n)}, t^{(n+k)}\right)}{\sigma^2},$$

onde σ^2 é a variância de $t^{(}n)$, e t é alguma função de interesse de θ.

- O Tamanho Efetivo da Amostra é dado por

$$N_{eff} = \frac{n}{1 + 2\sum_{k=1}^{\infty}\rho_k}.$$

Exemplo:

> Modelo Weibull:

No caso do modelo normal quando tem-se $X = (X_1, X_2, \ldots, X_n)$ uma amostra aleatória de $X \sim Weibull(\alpha, \gamma)$, temos a seguinte distribuição `a posteriori`

$$
\begin{aligned}
h(\alpha, \gamma \mid X) &\propto L(\alpha, \gamma \mid X) h(\alpha, \gamma) \\
&\propto L(\alpha, \gamma \mid X) h(\alpha) h(\gamma) \\
&\propto \alpha^n \gamma^n \left(\prod_{i=1}^{n} x_i^{\alpha - 1} \right) \exp\{-\gamma \textstyle\sum_{i=1}^{n} x_i^{\alpha}\} \times \frac{b_1^{a_1}}{\Gamma a_1} \alpha^{a_1 - 1} \exp\{-b_1 \alpha\} \\
&\times \frac{b_2^{a_2}}{\Gamma a_2} \alpha^{a_2 - 1} \exp\{-b_2 \gamma\}
\end{aligned}
$$

Exemplo:

> Modelo Weibull:

A distribuições condicionais completas são:

$$h(\alpha \mid \gamma, X) \propto \alpha^{n+a_1-1}\left(\prod_{i=1}^{n} x_i^{\alpha-1}\right)\exp\left\{-\gamma\sum_{i=1}^{n} x_i^{\alpha} - b_1\alpha\right\}$$

$$h(\gamma \mid \alpha, X) \propto \gamma^{n+a_2-1}\exp\left\{-\gamma\left(\sum_{i=1}^{n} x_i^{\alpha} - b_2\right)\right\}$$

Logo,

$(\gamma \mid \alpha, X) \sim Gama\left(n + a_2, \sum_{i=1}^{n} x_i^{\alpha} - b_2\right)$

$(\alpha \mid \gamma, X) \sim$???