Estatística computacional - Trabalho 3

Arthur Sonntag Kuchenbecker

Agosto 2019

1 Introdução

O objetivo do trabalho é comparar a construção de uma distribuição normal trivariada por meio (i) da decomposição $f(x,y,z) = f(x|y,z)f(y|z)f(z)$ com a construção utlizando o (ii) método de Gibbs Sampling.

Isso será feito por meio do software `R`, conforme explicado nas seções seguintes.

2 Método (i) decomposição teórica

Para a construção da distribuição com tamanho de amostra 5.000, foram utilizados os seguintes parâmetros.

$$\mu = \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix} \tag{1}$$

$$\Sigma = \begin{pmatrix} 4 & 2 & 3 \\ 2 & 9 & 4 \\ 3 & 4 & 16 \end{pmatrix} \tag{2}$$

Em que μ é o vetor com as médias das três variáveis (x, y, e z) e Σ é a matriz de covariâncias das mesmas.

Para encontrar a média e a variância das distribuições condicionais $f(y|z)$ e $f(x|y,z)$ foram utilizadas as seguintes relações.

$$f(y|z) \sim N(\bar{\mu}, \sqrt{\bar{\Sigma}}) \tag{3}$$

$$f(x|y,z) \sim N(\dot{\mu}, \sqrt{\dot{\Sigma}}) \tag{4}$$

Em que $\bar{\mu}$ e $\dot{\mu}$ representam as esperanças condicionais, e $\bar{\Sigma}$ e $\dot{\Sigma}$ representam as variâncias também condicionais.

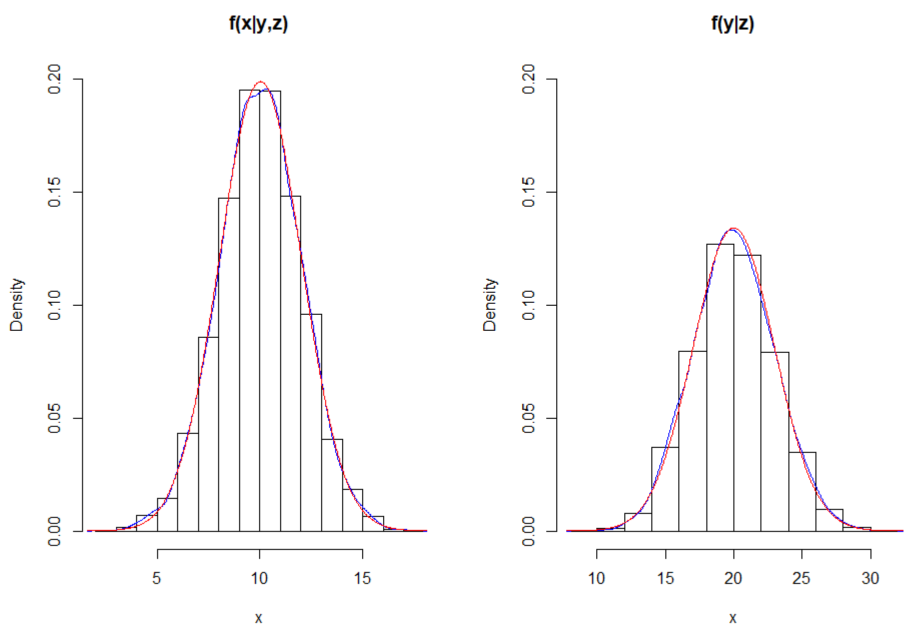
$$\bar{\mu} = \mu_y + \Sigma_{2,3}\Sigma_{3,3}^{-1}(z - \mu_z) \tag{5}$$

$$\bar{\Sigma} = \Sigma_{2,2} - \Sigma_{2,3}\Sigma_{3,3}^{-1}\Sigma_{3,2} \tag{6}$$

$$\dot{\mu} = \mu_x + \Sigma_{1,2:3}\Sigma_{2:3,2:3}^{-1}\begin{pmatrix} y - \mu_y \\ z - \mu_z \end{pmatrix} \tag{7}$$

$$\dot{\Sigma} = \Sigma_{1,1} - \Sigma_{1,2:3}\Sigma_{2:3,2:3}^{-1}\Sigma_{2:3,1} \tag{8}$$

Dado que a distribuição de z é normal, com média 30 e desvio-padrão 4, temos as seguintes distribuições para $f(y|z)$ e $f(x|y,z)$.



Pode-se perceber que ambas distribuições se assemelham muito a uma normal teórica com os parâmetros dados (linha em vermelho).

Além disso, ambas as distribuições apresentaram p-valor muito alto no teste "Kolmogorov-Smirnov", indicando que não podemos rejeitar a hipótese nula de que possuem distribuição normal.

```
> # Testando a hipotese de normalidade
> ks.test(x[,1],"pnorm",mean(x[,1]),sd(x[,1]))

        One-sample Kolmogorov-Smirnov test

data:  x[, 1]
```

```
D = 0.007107, p-value = 0.9623
alternative hypothesis: two-sided

> ks.test(x[,2],"pnorm",mean(x[,2]),sd(x[,2]))

        One-sample Kolmogorov-Smirnov test

data:  x[, 2]
D = 0.0075483, p-value = 0.9382
alternative hypothesis: two-sided
```

3 Método (ii) Gibbs Sampling

Gibbs Sampling é um algoritmo da classe dos métodos "Markov chain Monte Carlo (MCMC)" utilizado para obter sequências de observações que se aproximam de alguma distribuição multivariada, principalmente para casos em que a amostragem direta é difícil.

Neste exercício, o algoritmo será utilizado para encontrar a distribuição conjunta das varíaveis que compõem a distribuição normal multivariada, partindo das esperanças condicionais de cada uma delas e das suas variâncias também condicionais, calculados de maneira análoga ao demonstrado nas equações (05) até (08).

O chute dos valores iniciais para as três variáveis foram os seguintes:

```
# Chutando valores iniciais
x[1]=10
y[1]=20
z[1]=30
```

E a implementação do algoritmo se deu, basicamente, conforme código abaixo.

```
# Loop para criar a distribuicao
for (i in 2:n){
  mx_yz = mx+cx_xyxz%*%solve(cx_yz)%*%matrix(c(y[i-1]-my,z[i-1]-mz),c(2,1))
  # f(x|y,z)
  x[i] = rnorm(1,mx_yz,dx_yz^.5)
  my_xz = my+cy_yxyz%*%solve(cy_xz)%*%matrix(c(x[i]-mx,z[i-1]-mz),c(2,1))
  # f(y|x,z)
  y[i] = rnorm(1,my_xz,dy_xz^.5)
  mz_xy = mz+cz_zxzy%*%solve(cz_xy)%*%matrix(c(x[i]-mx,y[i]-my),c(2,1))
  # f(z|x,y)
  z[i] = rnorm(1,mz_xy,dz_xy^.5)

}
```

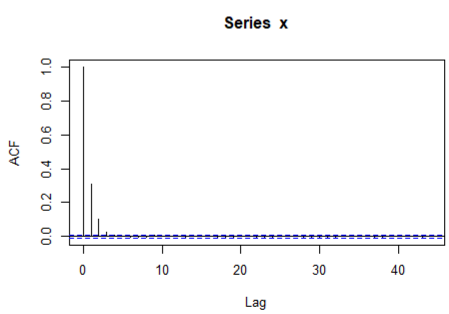
```
# Tratamento para excluir as primeiras amostras e pegar valores espacados,
# de forma a quebrar a possivel estrutura de autocorrelacao
xc = yc = zc = rep(0,nf)

for (i in 1:nf){
  xc[i] = x[(bi+1)+(i-1)*de]
  yc[i] = y[(bi+1)+(i-1)*de]
  zc[i] = z[(bi+1)+(i-1)*de]

}
```

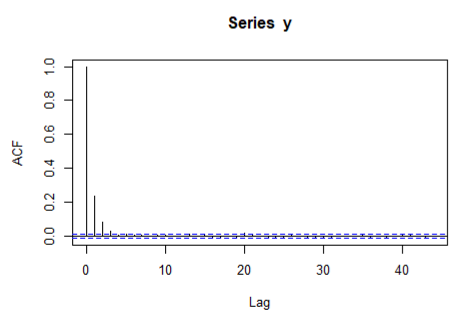
Como acontece com outros algoritmos da classe MCMC, o Gibbs Sampling gera amostras de uma cadeia de Markov, o que significa que cada uma delas está correlacionada, pelo menos, com a amostra imediatamente anterior.

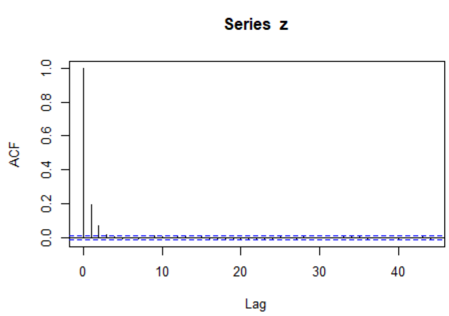
Por causa disso, para garantir a independência das variáveis, são descartadas as primeiras amostras, e são utilizadas janelas amostrais para construir a distribuição final.

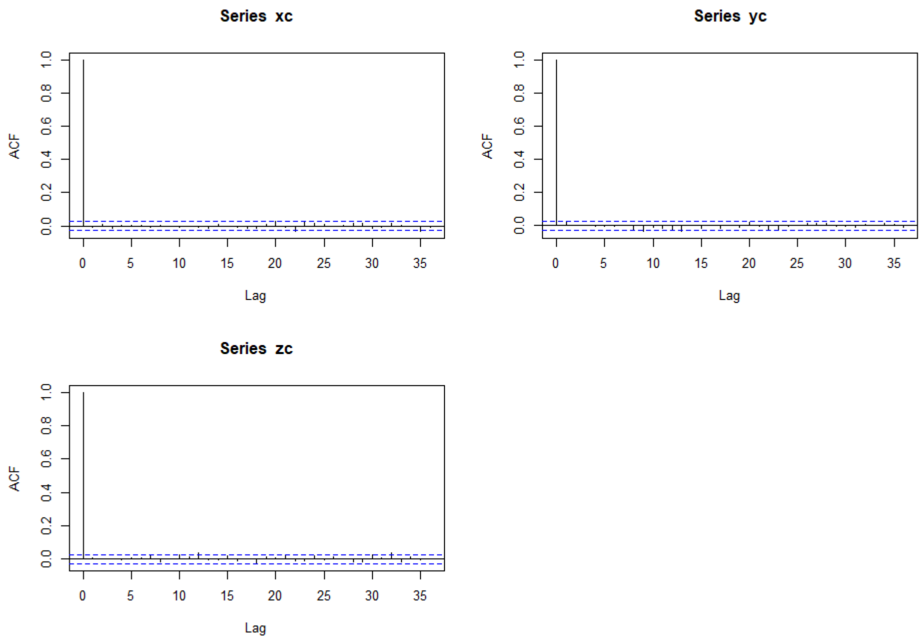
Para este exercício, como o objetivo era construir uma distribuição multivariada com tamanho de amostra 5.000, inicialmente foram gerados 30.000 valores, sendo que os 5.000 primeiros foram descartados, e a janela de amostras consideradas foi de 5 em 5.

Abaixo seguem os plots com as estimativas de autocorrelação antes e depois do tratamento realizado.

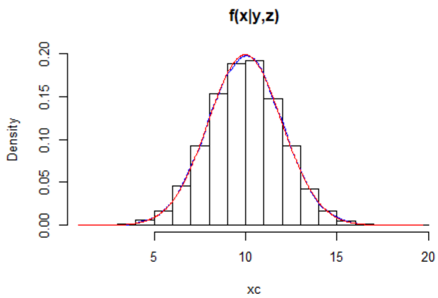


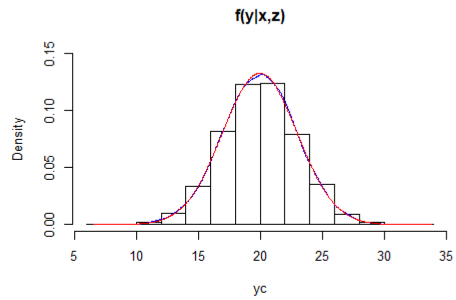


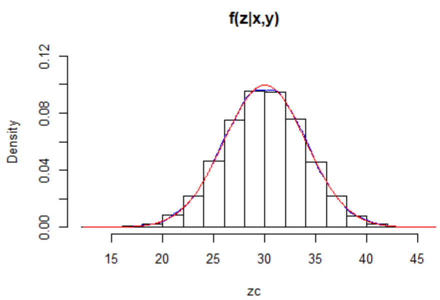



Series xc
Series yc
Series zc
ACF
Lag
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
30
35

Por fim, seguem abaixo os gráficos com as distribuições de x, y e z geradas pelo algoritmo, bem como o resultado do mesmo teste "Kolmogorov-Smirnov", indicando (novamente) que não pode-se rejeitar a hipótese nula de que as três variáveis possuem distribuição normal.








```
> # Testando a hipotese de normalidade
> ks.test(xyz[,1],"pnorm",mean(xyz[,1]),sd(xyz[,1]))

        One-sample Kolmogorov-Smirnov test

data:  xyz[, 1]
D = 0.0028419, p-value = 0.9687
alternative hypothesis: two-sided

> ks.test(xyz[,2],"pnorm",mean(xyz[,2]),sd(xyz[,2]))

        One-sample Kolmogorov-Smirnov test

data:  xyz[, 2]
D = 0.0029098, p-value = 0.9613
alternative hypothesis: two-sided

> ks.test(xyz[,3],"pnorm",mean(xyz[,3]),sd(xyz[,3]))

        One-sample Kolmogorov-Smirnov test
```

```
data:  xyz[, 3]
D = 0.0033746, p-value = 0.8841
alternative hypothesis: two-sided
```