DISTRIBUIÇÃO POSTERIOR MULTIVARIADA COM APROXIMAÇÃO GAUSSIANA USANDO RECURSOS DO R

PAUL GERHARD KINAS[1], MERHY HELI PAIVA RODRIGUES[2]

Resumo **–** Na inferência estatística bayesiana a distribuição de probabilidade posterior de parâmetros desconhecidos dos modelos é de importância central. Este artigo utiliza ferramentas da linguagem do R para obter uma amostra simulada da distribuição posterior bem como sua aproximação por meio de uma distribuição Gaussiana multivariada. O método é ilustrado com inferência para a curva de crescimento de Schnute e aplicado para as toninhas (*Pontoporia blainvillei*) do sul do Brasil.

Palavras chaves: inferência bayesiana, R, modelo de crescimento, distribuição Gaussiana multivariada.

MULTIVARIATE POSTERIOR DISTRIBUTION WITH GAUSSIAN APPROXIMATION, USING R.

Fundação Universidade do Rio Grande - FURG
[1] Ph. D. em Estatística - paulkinas@furg.br
[2] Graduada em Matemática Licenciatura - merhyheli@hotmail.com

***Abstract* –** In bayesian statistical inference the posterior probability distribution of unknown model parameters is of central importance. This paper uses the tools of the R language to obtain simulated posterior samples as an approximated Gaussian multivariate distribution. The method is illustrated with inference for the Schnute growth curve and applied to the fransciscana dolphin (*Pontoporia blainvillei*) from southern Brazil.

Key words: bayesian inference, R, growth model, Gaussian multivariate distribution.

I. INTRODUÇÃO

No enfoque bayesiano a probabilidade de um evento é definida como a plausibilidade da sua veracidade. Esta formalização de probabilidade [1] [2], como uma métrica de lógica indutiva, é mais abrangente que a axiomatização de Kolmogorov utilizada na estatística convencional, estendendo consideravelmente as possibilidades de aplicação. Na inferência estatística bayesiana, as informações disponíveis em dados y são modelados pela *verossimilhança* $p(y|\theta)$ parametrizada por θ, são acrescidas com as incertezas sobre θ que, por sua vez, são modeladas com uma distribuição de probabilidade denominada *priori* simbolizada por $p(\theta)$.

Os dois componentes de informação são combinados com o auxílio do Teorema de Bayes, resultando na *distribuição posterior para o parâmetro*, $p(\theta|y)$, e definida pela fórmula:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

A distribuição posterior para θ é o elemento central da inferência bayesiana e soluções analíticas nem sempre existem. Nestes casos recorre-se a simulações estocásticas para obter uma amostra empírica dessa distribuição. No entanto, quando θ é multidimensional, a eficiência dos algoritmos de simulação se torna fator limitante.

O objetivo do presente trabalho é ilustrar um método para obter uma

amostra posterior de θ utilizando uma distribuição gaussiana multivariada como distribuição posterior aproximada. A ferramenta computacional é a linguagem R [3]. Essa solução aproximada muitas vezes já é satisfatória. Em outros casos ela serve como primeiro passo para procedimentos mais elaborados. [4]

II. MÉTODO

O procedimento tem duas fases: (1) encontrar o vetor de médias e a matriz de covariância que caracterizem a distribuição gaussiana multivariada; (2) obter uma amostra simulada de θ desta distribuição.

A média será o valor de θ que maximiza $p(\theta|y)$. Encontrar esse valor equivale a minimizar a função

$$-\log p(\theta|y) = -\log p(y|\theta) - \log p(\theta) + k$$

Onde k é uma constante irrelevante na minimização.

Para um conjunto de dados $y = \{y_i; i = 1, \cdots, n\}$ que segue uma densidade de probabilidade definida por uma distribuição qualquer $f(y_i|\theta)$ tem-se que

$$-\log p(y|\theta) = -\sum_{i=1}^{n} \log f(y_i|\theta).$$

A maximização é efetuada no R com a função de otimização `optim`, baseada em Nelder–Mead e que usa algoritmos de gradiente conjugados quasi-Newton. A função necessita receber parâmetros de entrada `(par =)` a função para minimização `(fn =)` o método escolhido `(method =)` e permite a opção de produzir também a matriz hessiana de segundas derivadas parciais `(hessian =)`. A inversa dessa matriz é uma aproximação para a matriz de covariância que será necessária na próxima etapa.

Uma vez conhecido o máximo $\hat{\theta}$ e a matriz hessiana H, recorremos à função `mvrnorm` da biblioteca MASS, que produz amostras simuladas da distribuição multivariada normal. Essa função recebe o número de amostras a serem geradas (m), a média (mu) e a matriz de covariância (cov).

III. RESULTADOS

Para ilustrar o procedimento, obtemos amostras simuladas da distribuição posterior dos parâmetros do modelo de crescimento de Schnute [5], aplicado aos dados de idade e comprimento da toninha (Pontoporia blainvillei) extraídos de [6], separados por sexo.

A equação genérica do modelo de crescimento de Schnute é escrita a seguir para o tamanho y_i de um animal de idade t_i. Supõe-se uma distribuição log-Normal $y_i \sim logN(\mu_i, \sigma)$, denotando $z_i \sim N(0,1)$ uma variável gaussiana padronizada. Os valores $(\tau_1 < \tau_2)$ são duas idades arbitrariamente fixadas previamente e para as quais (γ_1, γ_2) são os tamanhos médios desconhecidos. Os parâmetros a e b definem o formato da curva de crescimento. Por exemplo, quando $a > 0$ a curva tem ponto de inflexão; se além disso $b = 0$, resulta o modelo de Gompertz, etc.

$$y_i = \left[\gamma_1^b + (\gamma_2^b - \gamma_1^b) \cdot \frac{1-e^{-a(t_i-\tau_1)}}{1-e^{-a(\tau_2-\tau_1)}}\right]^{1/b} \cdot e^{z_i\sigma} = e^{(\mu_i + z_i\sigma)}$$

Portanto, para modelo genérico de Schnute, o parâmetro $\theta = (a, b, \log\gamma_1, \log\gamma_2, \log\sigma)$ é um vetor de 5 elementos. Os parâmetros que são estritamente positivos foram log-transformados para melhor eficiência do algoritmo de otimização.

Supondo uma distribuição *priori* não-informativa de Jeffreys [7], isto é, $p(\theta) = \sigma^{-2}$, a função para minimização será

$$-\log(p\theta|y) = -\sum_{i=1}^{n} \log f_{lN}(y_i|\mu_i, \sigma) - 2\log(\sigma)$$

Onde $f_{lN}(\cdot)$ denota uma densidade log-Normal.

Os pontos iniciais utilizados para toninhas machos e fêmeas, foram par = $(1, 1, \log\overline{y}_1, \log\overline{y}_2, 0)$ sendo $\overline{y}_i$ definido como o tamanho médio dos indivíduos de idade t_i na amostra. Os sexos foram analisados separadamente. Com a otimização dos parâmetros, chegou-se às estimativas listadas nas colunas "moda" da Tabela I. Abaixo estão às matrizes de covariância que são obtidas invertendo as matrizes hessianas usando a função solve do R:

Matriz de Covariância das Fêmeas

a	b	γ_1	γ_2	σ
0.03875541 5	− 0.498633029	0.000446318	− 0.005363119	0.000000029
− 0.498633029	7.084866977	− 0.012017114	0.053776661	− 0.000000353
0.000446318	− 0.012017114	0.001091595	− 0.000019964	0.000000001
− 0.005363119	0.053776661	− 0.000019964	0.001223075	− .0000000002
0.000000029	− 0.000000353	0.000000001	− .0000000002	0.008474606

Matriz de Covariância dos Machos

a	b	γ_1	γ_2	σ
0.099919	−1.1556933	0.0007915	−0.0072093	0.0000025
−1.1556933	14.7022072	−0.0184452	0.0654331	−0.0000353
−0.0007915	−0.0184452	0.001509	−0.0000224	−0.0000001
−0.0072093	0.0654331	−0.0000224	0.0008627	−.00000001
0.0000025	−0.0000353	−0.0000001	−0.0000001	0.0087718

A distribuição posterior para γ_1 de machos e fêmeas pode ser visualizada na Figura 1. O gráfico de contorno da distribuição conjunta para (a, b) tanto para fêmeas quanto para machos estão representadas na Figura 2 (a sintaxe para a função que produz os contornos está no Anexo 1).

Tabela I: Moda (mo), Média (me) e Desvio Padrão (dp) das Distribuições Posteriores Marginais dos 5 Parâmetros do Modelo de Schnute, para toninha (*Pontoporia blainvillei*) Separado por Sexo.

	Fêmeas			Machos		
	moda	média	dp	Moda	média	dp
a	0.15	0.15	0.19	0.26	0.25	0.32
b	5.5	5.47	2.65	6.22	6.28	3.83
γ_1	89.2	89.26	2.95	83.71	83.69	3.25
γ_2	157.04	157.21	5.41	136.48	136.62	4.02
σ	0.07	0.07	0.007	0.07	0.07	0.006

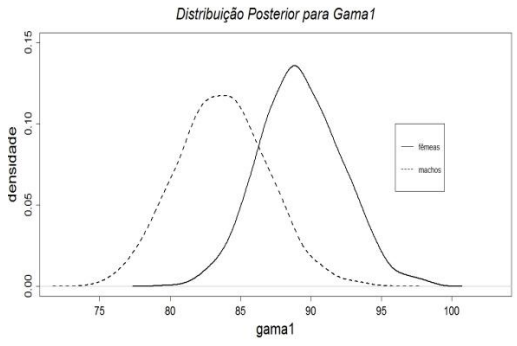

FIGURA 1: Distribuição posterior para o tamanho médio populacional (γ_1) de toninhas (*Pontoporia blainvillei*) com idade $\tau_1 = 0$, separados por sexo.

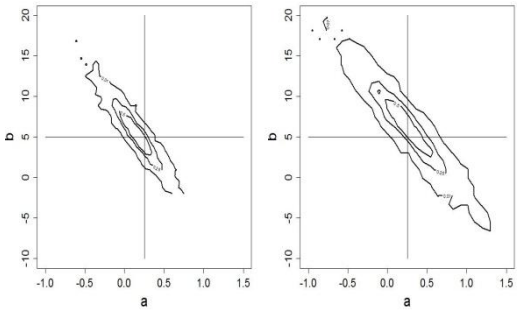

FIGURA. 2: Distribuição posterior para (a, b) com gráficos de contorno para as amostra simulada. Fêmeas (esquerda) e machos (direita). Linhas de contorno em 1%, 25% e 50% da altura máxima.

Com base nos parâmetros estimados verifica-se que machos tem tamanho médio menor que fêmeas já durante o primeiro ano de vida. Além disso, as curvas de crescimento são distintas, com machos crescendo rapidamente, mas sempre mantendo tamanhos médios menores que as fêmeas de mesma idade.

CONCLUSÃO

No presente trabalho, apresentamos uma aproximação gaussiana multivariada para o vetor de parâmetros do modelo de crescimento de Schnute. Embora o método envolva procedimento de otimização, inversão de matrizes e de simulação estocástica multivariada que são tecnicamente difíceis, a solução via linguagem R torna a sua operacionalização bastante facilitada.

Referências

[1] H. Jeffreys, *"Theory of Probability"*, 3rd ed., Oxford University Press, Oxford, UK, 1931.

[2] E.T, Jaynes, "*Probability Theory: The Logic of Science"*. New York, Cambridge University Press, 2003, 727p.

[3] R Development Core Team, R: A language and environment for statistical computing, www.R-project.org, 2007.

[4] A. Gelman, J.B Carlin, H.S Stern and D.B Rubin, *"Bayesian Data Analysis"*, London, Chapman and Hall, 1995, 526p.

[5] J. Schnute, "*A versatile growth model with statiscally stable parameters"*, Can. J. Fish. Aquat. Sci 38: 1128 – 1140. 1981.

[6] T. Walter, *"Curvas de Crescimento aplicadas a organismos aquáticos. Um estudo de caso para toninha Pontoporia blainvillei (Cetácea, Pontoporiide) do extremo sul do Brasil"*, Tese de Conclusão de Curso em Oceanologia, Fundação Universidade Federal do Rio Grande, Brasil, Dez. 1997.

[7] C. D. Paulino, M. A. Turkman and B. Murteira, "*Estatística Bayesiana",* Lisboa, Fundação Calouste Gulbenkian, 2003.

ANEXO 1

Função `sca.contour` para produzir os gráficos de contorno da figura 2.

```
sca.contour = function(var1,var2, n.classes=30,
        g.pc=0.1, q.line=c(0.01,0.25,0.5,0.80))
{ x.axis = var1
  y.axis = var2
  x.lim = c(min(x.axis),max(x.axis))
  y.lim = c(min(y.axis),max(y.axis))
  dif.x = x.lim[2]-x.lim[1]
  dif.y = y.lim[2]-y.lim[1]
```

```
  c.x = seq(x.lim[1]-g.pc*dif.x,x.lim[2]+
        g.pc*dif.x, length.out = n.classes)
  c.y = seq(y.lim[1]-g.pc*dif.y,y.lim[2]+
        g.pc*dif.y, length.out = n.classes)
  mp.x =  rep(0,n.classes-1)
  mp.y = mp.x
  for(i in 2:n.classes)
  { mp.x[i-1] =  mean(c.x[c(i-1,i)])
    mp.y[i-1] =  mean(c.y[c(i-1,i)])
  }
  p1.axis = cut(x.axis,breaks<-c.x,right=F)
  p2.axis = cut(y.axis,breaks<-c.y,right=F)
  f.axis = matrix(as.numeric(table(p1.axis,p2.axis)), ncol=
n.classes - 1 )
  f.axis = f.axis/max(f.axis)
  contour(mp.x,mp.y,f.axis,levels=q.line,
  lwd=2.0,add=F,xlab="a",ylab="b",main="")
}
```