Modelagem Bayesiana e Aplicações

Márcia D'Elia Branco

Universidade de São Paulo
Instituto de Matemática e Estatística
http:www.ime.usp.br/m̃branco

OpenBUGS e Coda

OpenBUGS

- O projeto **BUGS** (Bayesian inference using Gibbs sampling) foi desenvolvido por Spiegelhalter e Lauritzen no final da decada de 1980.
- Inicialmente, o mais comum era usar uma versão para o Window: **WinBUGS**.
- Em 2007 foi feita a última atualização do **WinBUGS**.
- Em 2004 Andrew Thomas iniciou o projeto **OpenBUGS**. Este projeto continua sendo atualizado até hoje.
- A interface com o **R** é realizada pelo pacote **R2OpenBUGS**.
- O projeto esta hospedado na *MRC Biostatistics Unit* da *University of Cambridge, UK*.

OpenBUGS

É necessário definir 4 tipos de entradas (*inputs*):

1. **Model:** um arquivo que descreve o programa na linguagem BUGS.
2. **Data:** o conjunto de dados.
3. **Parameters:** especifica o conjunto de parâmetros que se deseja monitorar durante a simulação.
4. **Initial values:** estabelece os valores inciais dos parâmetros do modelo, para estabelecer o ponto inicial da CM.

A saída (*output*) contem os valores simulados da CM e várias medidas resumos que podem ser utilizadas para inferência.

OpenBUGS

- Para o uso do pacote **R2OpenBUGS** é necessário primeiro baixar o programa OpenBUGS no computador.
- Em seguida, no **R**, baixar o pacote **R2OpenBUGS**.
- A estrutura geral da função *bugs* é dada por:

$$bugs(data, inits, parameters.to.save, "model.txt", \\ n.chains, n.iter, n.burnin, n.thin)$$

- Vamos ilustrar o uso com um problema simples de regressão linear normal :

$$y_i = b_0 + b_1 x_i + \epsilon_i \quad , \; \epsilon_i \sim N(0, 1/\tau)$$

Coda

- Teve inicio na decade de 1990 e desenvolvido junto com o projeto **BUGS**. Foi atualizado nos últimos anos, principalmente sua interface com o **R**.
- Realiza alguns diagnósticos de convergência e permite visualizar diversos gráficos .
- Tem função para obtenção do HPD.
- Obtenção das autocorrelações e da matriz de correlações entre os parâmetros.

1. Geweke: teste-t

- É baseada em apenas uma cadeia.
- Separa os valores simulados em duas amostras (após o *burn-in*). Uma de tamanho n_a no inicio da cadeia e outra de tamanho n_b no final.
- Realiza um teste de comparação de médias, usando a estatística:
$$Z = \frac{\bar{g}_a(\theta) - \bar{g}_b(\theta)}{(V_a + V_b)^{0.5}} \approx N(0, 1)$$
- Valores altos de $|Z|$ indicam não rejeição da hipótese de igualdade de médias.
- É uma técnica univariada, mas pode ser aplicada na função desvio $g(\theta) = -2ln(\pi(\theta))$.

2. Gelman-Rubin: $\hat{R}$

    - É baseada em multiplas cadeias. Em geral, um número pequeno de cadeias (2 ou 3) é suficiente.
    - Segue a ideia de análise de variância. Calcula a variância Entre as cadeias, B, e a variância Dentro das cadeias, W.
    - Determina a variância total V como uma média ponderada entre B e W.
    - A estatística é dada por

$$\hat{R} = \sqrt{\frac{V}{W}}$$

    - Esta estatística converge para 1 quando $M \to \infty$. Valores próximos de 1 indicam convergência.

3. Raftery-Lewis

- Assume que o interesse é estimar um determinado quantil da distribuição, q. É fixado um erro máximo r e uma probabilidade s da estimativa não se distanciar por mais de r.
- Com esses valores, busca estimar o número de interações N e o burn-in M.
- Denotamos por N_{min} um tamanho de amostra minimo (amostra piloto).
- A estatística é dada por

$$I = \frac{(M + N)}{N_{min}}$$

- Valores altos indicam alta influência dos valores iniciais.

Tamanho efetivo da amostra

- O tamanho efetivo da amostra ESS indica o tamanho de uma amostra independente da função de interesse $\pi(\theta)$, que resultaria em um estimador com o mesmo erro padrão (variância) do estimador obtido usando N simulações.
- O valor é dada por:

$$ESS = \frac{N}{1 + \sum_{k=1}^{\infty} \rho_k}$$

- Em que ρ_k é a autocorrelação de comprimento k

$$\rho_k = \frac{Cov(g^{(t)}(\theta), g^{(t+k)}(\theta))}{\sigma^2}$$