**Universidade de São Paulo
Escola Superior de Agricultura Luiz de Queiroz**

**Diretrizes para aplicação de inferência Bayesiana aproximada para
modelos lineares generalizados e dados georreferenciados**

Djair Durand Ramalho Frade

Tese apresentada para obtenção do título de Doutor em
Ciências. Área de concentração: Estatística e Experimen-
tação Agronômica

**Piracicaba
2018**

Djair Durand Ramalho Frade
Bacharel em Estatística

**Diretrizes para aplicação de inferência Bayesiana aproximada para
modelos lineares generalizados e dados georreferenciados**

versão revisada de acordo com a resolução CoPGr 6018 de 2011

Orientadora:

Profa. Dra. **SÔNIA MARIA DE STEFANO PIEDADE**

Tese apresentada para obtenção do título de Doutor em Ciências. Área de concentração: Estatística e Experimentação Agronômica

Piracicaba
2018

**Dados Internacionais de Catalogação na Publicação
DIVISÃO DE BIBLIOTECA - DIBD/ESALQ/USP**

Frade, Djair Durand Ramalho

Diretrizes para aplicação de inferência Bayesiana aproximada para modelos lineares generalizados e dados georreferenciados / Djair Durand Ramalho Frade. – – versão revisada de acordo com a resolução CoPGr 6018 de 2011. – – Piracicaba, 2018 .

63 p.

Tese (Doutorado) – – USP / Escola Superior de Agricultura “Luiz de Queiroz”.

1. INLA 2. Inferência bayesiana 3. Modelos lineares generalizados I Título

DEDICATÓRIA

Dedico ao meu amor Juliana da Silva Rodrigues

AGRADECIMENTOS

Gostaria de expressar minha gratidão a todos aqueles que me deram a possibilidade de concluir esta tese, especialmente minha mãe Darlene Durand Ramalho Frade e meus irmãos Darley Durand Ramalho Frade e Jordânia Durand Ramalho Frade, por seu amor e apoio durante toda a minha vida.

À minha namorada Juliana da Silva Rodrigues, pelo companheirismo desde nossa infância e pelo apoio ao longo de todo Doutorado. Muito obrigado por me entender.

A minha orientadora, Profa. Dra. Sônia Maria De Stefano Piedade, pelo apoio, pela paciência, motivação e por toda confiança.

A todos os meus amigos da ESALQ/USP Ana Cristina Alves Garcêz, Cayan Atreio Portela Bárcena Saavedra, Gustavo Thomas, Leomir Ferreira, Marcello Neiva de Mello, Pedro Marinho Amoedo, Pórtya Piscitelli Cavalcanti, Rafael Moral, Raquel Aline Oliveira Eloy, Reginaldo Francisco Hilário, Ricardo Klein Sercundes e Thiago de Paula Oliveira.

À equipe do Departamento de Ciências Exatas da ESALQ / USP, aos técnicos de informática Eduardo Bonilha e Jorge Wiendl em especial as secretárias Luciane Brajão e Solange Sabadin.

Este trabalho foi financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

SUMÁRIO

Resumo	6
Abstract	7
1 Introdução	9
Referências	11
2 Impacto de malhas não convexas em análise de dados geoestatísticos usando R-INLA	13
2.1 Introdução	14
2.2 INLA e SPDE	15
2.3 Seleção de modelos e medidas para avaliação do ajuste	17
2.3.1 Critério de informação deviance (DIC)	17
2.3.2 Critério de informação Watanabe-Akaike (WAIC)	18
2.3.3 Condicionais preditivas ordenadas (CPO)	19
2.4 Estudo de simulação	20
2.5 Conclusão	25
Referências	25
3 Impacto da escolha do método de aproximação para a distribuição condicional do campo aleatório nos modelos lineares generalizados	29
3.1 Introdução	30
3.2 INLA	31
3.2.1 Aproximação para $\pi(\boldsymbol{\theta} \mathbf{y})$	33
3.2.2 Aproximação para $\pi(x_i \boldsymbol{\theta}, \mathbf{y})$	34
3.3 Seleção de modelos e medidas para avaliação do ajuste	34
3.4 Estudos de simulação	36
3.5 Conclusão	40
Referências	41
Apêndices	43

RESUMO

Diretrizes para aplicação de inferência Bayesiana aproximada para modelos lineares generalizados e dados georreferenciados

Neste trabalho, exploramos e propusemos diretrizes para a análise de dados utilizando o método *Integrated Nested Laplace Approximation* - INLA para os modelos lineares generalizados (MLG's) e modelos baseados em dados georreferenciados. No caso dos MLG's, verificou-se o impacto do método de aproximação utilizado para aproximar a distribuição a posteriori conjunta. Nos dados georreferenciados, avaliou-se e propôs-se diretrizes para construção das malhas, passo imprescindível para obtenção de resultados mais precisos. Em ambos os casos, foram realizados estudos de simulação. Para selecionar os melhores modelos, foram calculadas medidas de concordância entre as observações e os valores ajustados pelos modelos, por exemplo, erro quadrático médio e taxa de cobertura.

Palavras-chave: INLA; Inferência bayesiana; Malhas; Modelos lineares generalizados

ABSTRACT

Approximate Bayesian inference guidelines for generalized linear models and georeferenced data

In this work, we explore and propose guidelines for data analysis using the Integrated Nested Laplace Approximation (INLA) method for generalized linear models (GLM) and models based on georeferenced data. In the case of GLMs, the impact of the approximation method used to approximate the a posteriori joint distribution was verified. In the georeferenced data, we evaluated and proposed guidelines for the construction of the meshes, an essential step for obtaining more precise results. In both cases, simulation studies were performed. To select the best models, agreement measures were calculated between observations and models, for example, mean square error and coverage rate. Keywords: INLA; Bayesian inference; Meshes; Generalized linear models

1 INTRODUÇÃO

Os métodos de Estatística Bayesiana estão ganhando uma maior visibilidade pelos pesquisadores, devido a sua simples estrutura e a sua alta aplicabilidade nas mais diversas áreas do conhecimento, além de possuírem interpretações mais intuitivas do que as da Estatística Clássica. A estimação Bayesiana necessita de programação computacional intensiva e os métodos mais difundidos para a obtenção de resumos da distribuição conjunta a *posteriori* são os Métodos de Monte Carlo Cadeias de Markov, MCMC, especialmente o algoritmo de Metropolis-Hasting e o amostrador de Gibbs (GEMAN; GEMAN, 1984).

A execução do algoritmo de Metropolis-Hasting e do amostrador de Gibbs em problemas complexos pode ser demorada para a obtenção da convergência. Uma alternativa para aproximar as distribuições marginais a *posteriori* foi proposta por Rue, Martino e Chopin (2009) baseada na abordagem INLA (Integrated Nested Laplace Approximation). A grande vantagem do método INLA é o seu tempo computacional significativamente menor que o tempo computacional de métodos de simulação, tais como MCMC, além de não ter a necessidade de verificação de convergência, já que o INLA é um método determinístico.

O INLA lida com modelos Gaussianos latentes (MGL), uma subclasse de modelos de regressão aditiva estruturada. Segundo Hickey (2015), esses modelos são formulados de forma muito similar aos modelos lineares generalizados (MLG's), mas o preditor linear é substituído por um preditor aditivo estruturado que pode incluir efeitos não-lineares. Em um MGL a variável resposta é assumida pertencer a uma família de distribuições (não necessariamente a família exponencial) em que algum parâmetro da família, μ_i , é ligado a um preditor aditivo estruturado, η_i , por meio de uma função de ligação $g(\cdot)$, de modo que $g(\mu_i) = \eta_i$. Este preditor contém todas as covariáveis e pode ser escrito como

$$\eta_i = \alpha + \sum_{j=1}^{n_f} f^{(j)}(u_{ji}) + \sum_{k=1}^{n_\beta} \beta_k z_{ki} + \varepsilon_i$$

As variáveis que compõem o preditor são um intercepto α , termos não lineares $f^{(j)}$, termos lineares β_k e um termo de erro ε_i . Portanto, α , β_k e ε_i podem ser entendidos de forma semelhante aos elementos do preditor de um MLG. Os termos adicionais não lineares $f^{(j)}$ podem assumir diferentes formas, como efeitos não-lineares de covariáveis, tendências temporais e efeitos sazonais, intercepções aleatórias, bem como efeitos aleatórios temporais ou espaciais.

Conforme Rue, Martino e Chopin (2009), usando a mesma notação, a classe de

modelos Gaussianos latentes, definidos em três estágios, é especificada por

$$\begin{aligned} \mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_1 &\sim \pi(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_1), & (\text{Equação de observações}) \\ \mathbf{x}|\boldsymbol{\theta}_2, \mathbf{y} &\sim N(\boldsymbol{\mu}(\boldsymbol{\theta}_2), \mathbf{Q}^{-1}(\boldsymbol{\theta}_2)), & (\text{Campo Gaussiano Latente}) \\ \boldsymbol{\theta} &\sim \pi(\boldsymbol{\theta}), & (\text{Hiperparâmetros}) \end{aligned}$$

em que $\mathbf{Q}(\boldsymbol{\theta}_2)$ é a matriz de precisão do processo Gaussiano do vetor $\mathbf{x}$. Para que o modelo seja válido, a matriz de variâncias tem que ser simétrica e positiva definida.

O INLA consiste em construir aproximações de Laplace, em que originalmente se aproximava as distribuições *a posteriori*. O primeiro passo na aproximação INLA é realizar uma aproximação de Laplace para a *a posteriori* conjunta

$$\pi(\boldsymbol{\theta}|\mathbf{y}) \propto \frac{\pi(\mathbf{x}, \boldsymbol{\theta}, \mathbf{y})}{\tilde{\pi}_G(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})},$$

em que $\tilde{\pi}_G(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ é a aproximação Gaussiana para $\pi(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$, que corresponde à verdadeira distribuição da moda (RUE; MARTINO; CHOPIN, 2009). Essa aproximação, segundo Martino e Rue (2007), é equivalente à aproximação de Laplace devido a Tierney e Kadane (1986) para uma distribuição marginal *a posteriori*.

No segundo passo, para aproximar a distribuição marginal *a posteriori* de cada um dos elementos de $\mathbf{x}$, fixados valores de $\boldsymbol{\theta}$, três métodos podem ser considerados: aproximação gaussiana (que no contexto dos GMRF, é chamada de aproximação GMRF), aproximação de Laplace e aproximação simplificada de Laplace. A diferença entre os três métodos varia em precisão e custo computacional.

O INLA pode ser combinado com a abordagem de Equação Diferencial Parcial Estocástica (Stochastic Partial Differential Equation - SPDE) (Lindgren, Rue e Lindström, 2011) para modelar de forma eficaz aplicações em dados georreferenciados (BIVAND; GOMEZ-RUBIO; RUE, 2015). Quando usado em conjunto com SPDE para modelos geostatísticos, é necessário gerar uma malha, formada por junções de triângulos, sob a região de estudo. A precisão das estimativas dos parâmetros vai depender dessa malha. Um conjunto de atribuições precisa ser cuidadosamente escolhido para criar uma malha adequada: definir valores para as bordas dos triângulos dentro da malha (que engloba a região de estudo), para as bordas da malha externa (fora da região de estudo, para evitar o efeito de borda) e para a menor distância permitida entre as junções das arestas (nós). A utilização desses atributos implica em malhas com uma cobertura convexa. O INLA permite criar malhas com cobertura não convexa por meio da função `inla.nonconvex.hull(..., convex, resolution)`. Os argumentos *convex* e *resolution* definem os valores da extensão do raio de cada triângulo e a resolução interna para o cálculo, respectivamente. Cada mudança nesses valores implica mais ou menos nós na malha, o que implica mais ou menos carga computacional. Neste trabalho detonaremos o raio interno por a e a resolução interna

por b . Diferentes combinações desses dois argumentos geram diferentes malhas.

Os principais objetivos deste trabalho são encontrar uma maneira de criar a malha mais adequada para dados georreferenciados, por meio de um estudo de simulação, e verificar o impacto de método de aproximação para a distribuição condicional do campo aleatório nos modelos lineares generalizados.

Referências

BIVAND, R.; GOMEZ-RUBIO, V.; RUE, H. Spatial data analysis with r-inla with some extensions. *J. Stat. Softw*, v. 63, n. 20, p. 1–31, 2015.

GEMAN, S.; GEMAN, D. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Pattern Anal. Mach. Intell*, Springer, v. 6, p. 721–741, 1984.

HICKETIER, A. *Spatio-temporal Modeling of Hantavirus in Germany*. 2015. 45p. Tese (Doutorado em Estatística Matemática) - Mathematical Statistics, Stockholm University, Piracicaba, 2017.

LINDGREN, F.; RUE, H.; LINDSTRÔM, J. An explicit link between Gaussian fields and Gaussian Markov random fields: the SPDE approach (with discussion). *J. R. Stat. Soc. Ser. B.*, v. 73, p. 423–498.

MARTINO, S.; RUE, H. Approximate bayesian inference for hierarchical gaussian Markov random fields models. *Journal of Statistical Planning and Inference*, Wiley Online Library, v. 137, p. 3177–3192, 2007.

RUE, H.; HELD, L. *Gaussian Markov Random Fields: Theory and Applications*. Chapman & Hall, 2005. 280p.

RUE, H.; MARTINO, S.; CHOPIN, N. Appriximate Bayesian inference for latent Gaussian models by using integrated nested Laplace aproximations (with discussion). *Journal of the Royal Statistical Society–Series B*, Wiley Online Library, v. 71, Part 2, p. 319–392, 2009.

TIERNEY, L.; KADANE, J. B. Accurate approximations for posterior momentos and marginal densities. *Journal of the American Statistical Association*, Wiley Online Library, v. 81, n. 393, p. 82–86, 1986.

2 IMPACTO DE MALHAS NÃO CONVEXAS EM ANÁLISE DE DADOS GEOESTATÍSTICOS USANDO R-INLA

Resumo: Estatísticas espaciais são úteis para medir propriedades e relações em um fenômeno geograficamente estudado. A maioria dos métodos usados nesta análise são computacionalmente exigentes e, portanto, o desenvolvimento de métodos mais eficazes em termos computacionais tem recebido muita atenção nos últimos anos. Uma abordagem que é considerada um desenvolvimento notável, é a introdução do método integrado de aproximação de Laplace aninhado que é capaz de realizar a análise Bayesiana de uma maneira mais eficiente. O uso deste método para dados geoestatísticos é comumente feito considerando a abordagem SPDE que requer a criação de uma malha sobre a área de estudo. Este é um passo importante na análise, pois os resultados obtidos dependem da escolha dessa malha. Aqui, o impacto da malha na inferência é investigado por meio de estudos de simulação. Como não há procedimento formal para especificar a malha, foram investigadas possíveis orientações sobre como uma malha ótima é escolhida. Por meio de um estudo de simulação, com um conjunto de dados aleatórios, diferentes malhas foram criadas, considerando os locais das amostras como ponto de partida para os nós da malha.

Palavras-chave: Malha; Monte Carlo; Bayesiana; Campo aleatório

Abstract: Spatial statistics are useful for measuring properties and relationships of a geographically studied phenomenon. Most of the methods used in this analysis are computationally demanding, and therefore the development of more computationally effective methods has received much attention in recent years. One approach, which is considered a remarkable development, is the introduction of the integrated nested Laplace approximation method that is able to perform Bayesian analysis in a more efficient way. The use of this method for geostatistical data is commonly done considering the SPDE approach that requires the creation of a mesh over the study area. This is an important step in the analysis, because the results obtained depend on the choice of this mesh. Here, the mesh impact on inference is investigated through simulation studies. Since there is no formal procedure for specifying the mesh, we investigate possible orientations about how an optimal mesh is chosen. Through a simulation study, with a random data set, different meshes were created, considering the locations of the samples as a starting point for the knots of the mesh.

Keywords: Mesh; Monte Carlo; Bayesian; Random field

2.1 Introdução

Nas últimas décadas, a disponibilidade de dados espaciais aumentou substancialmente, principalmente devido aos avanços em ferramentas computacionais que nos permitem coletar dados em tempo real provenientes de GPS, satélites, etc. Os métodos tradicionais utilizados para a análise desse tipo de dado foram desenvolvidos sem considerar a eficiência computacional, pois em muitas aplicações o tamanho da amostra foi limitado devido ao custo envolvido na obtenção das medidas (BOLIN; LINDGREN, 2013).

Os modelos geoestatísticos são amplamente utilizados para calcular a autocorrelação espacial para derivar previsões em locais não medidos (GELFAND *et al.*, 2010). Além das versões mais simples, esses modelos também podem ser estimados utilizando a abordagem bayesiana. No entanto, essa abordagem utiliza os métodos de amostragem como Monte Carlo Cadeia de Markov (MCMC), que demandam tempo de computação intenso. Como alternativa computacionalmente eficiente para o método MCMC, Rue, Martino e Chopin (2009) propuseram o uso de aproximações integradas de Laplace aninhadas (INLA). O INLA é projetado para modelos Gaussianos latentes, uma classe muito ampla e flexível de modelos que variam de modelos mistos lineares (generalizados) a modelos espaciais e espaço-temporais. Além disso, o INLA pode ser combinado com a abordagem da Equação Diferencial Parcial Estocástica (SPDE) proposta por (LINDGREN, 2012) a fim de implementar modelos espaciais e espaço-temporais para dados de referência de pontos.

A abordagem SPDE aproxima o campo Gaussiano usando um campo aleatório Gaussian Markov. Para dados geoestatísticos, o campo espacial é descrito usando uma soma ponderada de função de base linear por peça que é usualmente definida a partir de uma triangularização da área de estudo, chamada malha. A precisão das estimativas, bem como as previsões, dependerão da especificação da malha. A ideia é que triângulos menores aumentem a precisão, embora quando o alcance espacial for maior, triângulos maiores podem ser usados. A malha pode ser construída usando diferentes princípios. Um método comum, usando R-INLA, consiste em usar as localizações das medidas como vértices iniciais de triangulação e adicionar ou remover nós de acordo com o tamanho das bordas dos triângulos da malha interna (que inclui a região de estudo), para as bordas da malha externa (fora da área de estudo, para evitar efeitos de borda) e a menor distância permitida entre as junções das bordas (nós). Para evitar o efeito de borda, o R-INLA fornece a opção de calcular uma cobertura não convexa para usar como informações no ajuste.

Como a análise depende da malha escolhida, o objetivo deste trabalho é estudar extensivamente os parâmetros estimados para a criação de diretrizes para confecção das malhas utilizando o efeito de bordas não convexas, resolvendo algumas questões levantadas por Righetto (2017). Na Seção 2.2, apresentamos uma breve introdução ao INLA e à abordagem do SPDE e discutimos alguns critérios para avaliar a malha. Um estudo de

simulação é realizado na Seção 2.4.

2.2 INLA e SPDE

Considere um campo aleatório $\mathbf{X}$ com realizações nos locais s_i , $i = 1, \dots, n$: $X(s_i)$ que são normalmente distribuídos com média 0 e covariância conjunta que pode ser representada como uma função de covariância Matérn:

$$\text{Cov}(X(s_i), X(s_j)) = \sigma_e^2 \delta_{ij} + \frac{\sigma_x^2}{2^{\nu-1} \Gamma(\nu)} (\kappa \|s_i - s_j\|)^\nu K_\nu(\kappa \|s_i - s_j\|), \quad (2.1)$$

em que $\|s_i - s_j\|$ é a distância Euclidiana entre os pontos s_i e s_j , σ_x^2 é a variância, κ é um parâmetro de escala, ν parâmetro de suavização, K_ν é a função de Bessel modificada de segunda ordem, σ_e^2 é o ruído branco, δ_{ij} é o delta de Kronecker: $\delta_{ij} = 0$ quando $i = j$ e $\delta_{ij} = 1$ quando $i \neq j$, e Γ é a função Gama (MINASNY; MCBRATNEY, 2013).

A função de covariância de Matérn é amplamente utilizada em vários campos científicos, mas a relação importante que devemos considerar, segundo Lindgren, Rue e Lindström (2011), é que um campo aleatório $x(\mathbf{s})$ com a função de covariância de Matérn é uma solução linear fracionada de uma equação diferencial estocástica parcial (SPDE, do inglês *Stochastic Partial Differential Equations*), conforme denotado por:

$$(\kappa^2 - \Delta)^{\omega/2} \mathbf{X}(\mathbf{s}) = \mathbf{W}(\mathbf{s}), \quad (2.2)$$

em que $\mathbf{W}(\mathbf{s})$ é o ruído branco, Δ é o operador de Laplace $\delta^2/\delta_{s_i^2} + \delta^2/\delta_{s_j^2}$ (s_i e s_j representam as coordenadas no espaço bi-dimensional), ω é um inteiro positivo relacionado com o parâmetro de suavização ν , em que $\omega = \nu + 1$ (LINDGREN; RUE, 2015).

Lindgren, Rue e Lindström (2011) mostram que uma aproximação à solução do SPDE definido em (2.2) pode ser obtida usando um método de elementos finitos (FEM, do inglês *Finite Element Method*), uma técnica numérica para resolver equações diferenciais parciais. A solução aproximada é computacionalmente eficiente para fins inferenciais. O primeiro passo é substituir o campo aleatório dimensional infinito por uma representação da função de base finita por meio da triangulação do domínio:

$$\mathbf{X}(\mathbf{s}) = \sum_{k=1}^K \psi_k(\mathbf{s}) \mathbf{x}_k, \quad (2.3)$$

em que $\mathbf{x}_k$ são pesos gaussianos escolhidos para aproximar as soluções para o SPDE no domínio e ψ_k são funções de base linear definidas na triangulação do domínio, ou seja, uma subdivisão em triângulos sem interseção com K vértices. Uma interpretação da representação (2.3) com a escolha de funções de base é que os pesos determinam os valores do campo nos vértices, e os valores no interior dos triângulos são determinados por in-

interpolação linear. A distribuição total da solução continuamente indexada é determinada pela distribuição conjunta dos pesos (LINDGREN; RUE; LINDSTRÖM, 2011).

Uma ilustração da abordagem SPDE é apresentada na Figura 2.1, que exibe um campo aleatório espacial contínuo e a representação de elemento finito correspondente, com funções de base linear por partes sobre uma malha triangulada.

Figura 2.1: Exemplo de um campo aleatório contínuo e a representação de elemento finito correspondente com funções de base linear por partes sobre uma malha triangulada.

O primeiro passo da análise é a construção de uma malha triangular na qual a representação SPDE é construída (LINDGREN, 2012). Como passo inicial, podemos optar por colocar os vértices do triângulo nos locais onde as medições foram feitas e, depois disso, adicionar ou remover vértices adicionais, criando a triangulação para a análise. Nos casos mais comuns, considera-se que os locais de observação são espalhados semi-aleatoriamente em uma região do espaço tal que não haja limite físico, apenas uma região de observação limitada. Ao lidar apenas com covariâncias entre pontos de dados, esta distinção é muitas vezes sem importância, mas aqui se torna uma parte possivelmente vital do estudo, já que o SPDE exibirá efeitos de borda. Segundo LINDGREN (2012), na implementação do R-INLA, são utilizados os limites de Neumann, o que aumenta a variância perto da borda. Ainda, segundo o autor, o efeito geral da construção da triangulação é que pode-se obter triângulos pequenos (e assim obter alta precisão para representar o campo) onde os locais são densos, e maiores quando os locais são escassos (e, portanto, fornecem informações menos detalhadas) e grandes triângulos onde não há observações, ocasionando um gasto computacional desnecessário. Assim, como a análise depende da malha, é muito importante criar uma malha adequada para os dados disponíveis.

Na implementação do R-INLA, a função *inla.mesh.2d()* está disponível para criar a malha. Vários parâmetros nesta função podem ser escolhidos de modo a construir uma malha para o problema em questão, por exemplo, os parâmetros *max.edge* definem os comprimentos máximos permitidos de arestas triangulares no domínio interno e na extensão externa, o parâmetro de corte corresponde à menor distância permitida entre os pontos. Estes parâmetros foram exhaustivamente estudados por Righetto (2017), no en-

tanto, todas as malhas estudadas pela autora foram feitas para ter uma borda do domínio convexa. Pode-se construir bordas não convexas usando a função *inla.nonconvex.hull()*. Nesta função é passado os pontos e definindo alguma restrição. Podemos controlar a forma da borda, incluindo sua convexidade, por meio do parâmetro *convex* e a resolução interna com o argumento *resolution*.

Trabalha-se com os dois argumentos da função *inla.nonconvex.hull()* para construir as malhas não convexas: a concavidade (*a*) e a resolução (*b*). Os valores considerados para a concavidade correspondem a 7%, 15%, 25% e 35% da maior distância Euclidiana observada entre dois pontos observados na região de estudo. Para a resolução, os valores considerados foram 25, 40 e 60.

Por razões exploratórias, ilustrou-se como a escolha destes parâmetros determina o aspecto da malha na Figura 2.2. Considerou-se 100 locais gerados aleatoriamente de um processo espacial no quadrado unitário. Começando com esses locais como vértices iniciais da triangulação, criando diferentes malhas, dependendo dos parâmetros *a* e *b*. As malhas construídas em 2.2a, 2.2b e 2.2c foram geradas fixando o parâmetro *convex* (7%) e variando o parâmetro *resolution* (25, 40 e 60). Com o aumento do parâmetro *resolution*, a triangulação se torna mais densa, com o número de vértices aumentando de 959 para 987 e 1020, respectivamente. As malhas apresentadas nas Figuras 2.2d, 2.2e e 2.2f foram geradas fixando o parâmetro *resolution*, variando o parâmetro *convex* (15%, 25% e 35%). Com o aumento do parâmetro *convex*, o número de vértices aumenta de 1398 para 1899 e 2510, respectivamente. Mesmo pequenas mudanças em um parâmetro para construir a malha podem gerar uma malha completamente diferente e os resultados podem ser afetados. Então, o foco desta pesquisa é investigar o impacto da malha na análise.

2.3 Seleção de modelos e medidas para avaliação do ajuste

Para avaliar e comparar o desempenho dos modelos, diferentes critérios de informação preditiva podem ser usados. Ghosh *et al.* (2009), apresentaram três abordagens para seleção Bayesiana de modelos: a primeira é o Critério de Informação Deviance (*Deviance Information Criterion* - DIC) (Spiegelhalter *et al.* (2000)), o critério de informação de Watanabe-Akaike (WAIC) e as condicionais preditivas ordenadas (*Conditional predictive ordinate* - CPO).

2.3.1 Critério de informação deviance (DIC)

Para o cálculo do DIC considera-se a medida *deviance* dada por $D = -2 \log p(\mathbf{y}|\hat{\boldsymbol{\theta}})$, em que $\mathbf{y}$ é o vetor de dados, $\boldsymbol{\theta}$ é o vetor de parâmetros de interesse sob determinado modelo e $p(\mathbf{y}|\boldsymbol{\theta})$ a função de verossimilhança. O DIC é calculada como $DIC = \hat{D} + 2p_D$, em que $\hat{D} = E_{\boldsymbol{\theta}|\mathbf{y}}[D]$ é uma medida que indica a qualidade do ajuste, e p_D o número efetivo de parâmetros, representando a complexidade do modelo. Segundo Spiegelhalter

Figura 2.2: Exemplos de malhas considerando diferentes valores para os parâmetros a e b

et al. (2000), a quantidade efetiva de parâmetros do modelo é calculado como a média a *posteriori* da *deviance* menos a *deviance* avaliada na média a *posteriori*, ou seja, $p_D = \hat{D} - D(\hat{\theta})$.

A ideia é que os modelos com menor DIC devem ser preferidos a modelos com maiores valores para essa estatística. Os modelos são penalizados tanto pelo valor de $\hat{D}$, o que favorece um bom ajuste, como também pelo número efetivo de parâmetros p_D .

É importante ressaltar que para o cálculo do DIC assume-se que a média a *posteriori* é uma boa estimativa pontual para os parâmetros e, portanto, não representa uma boa medida para a avaliação do ajuste na presença de distribuições a *posteriori* assimétricas para os parâmetros do modelo.

2.3.2 Critério de informação Watanabe-Akaike (WAIC)

O critério WAIC foi proposto por Watanabe (2010) e segundo Gelman, Hwang e Vehtari (2013), é visto como uma melhoria do critério DIC. O WAIC é calculado por

$$WAIC = -2 \log \int p(\mathbf{y}|\boldsymbol{\theta}) \prod (\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} + \sum_{i=1}^n Var(\log(p(y_i|\boldsymbol{\theta}))). \quad (2.4)$$

O primeiro termo mede o desempenho preditivo, e o segundo termo penaliza a complexidade do modelo. Segundo Gelman, Hwang e Vehtari (2013), o termo complexidade é uma soma da variância para cada ponto de dados separadamente, gerando resultados mais estáveis como DIC.

2.3.3 Condicionais preditivas ordenadas (CPO)

A estatística CPO é um método de avaliação cruzada utilizado para avaliar a performance do modelo medindo o seu poder preditivo (GEISSER; EDDY, 1979). A distribuição preditiva a *posteriori* para a i -ésima observação é dada por

$$p(\mathbf{y}_i|\mathbf{y}_{-i}) = \int p(\mathbf{y}_i|\boldsymbol{\theta}) \prod (\boldsymbol{\theta}|\mathbf{y}_{-i}) d\boldsymbol{\theta},$$

em que $\mathbf{y}_i$ é a i -ésima observação de $\mathbf{y}$ e $\mathbf{y}_{-i}$ é $\mathbf{y}$ sem a observação i . Então, o CPO é a probabilidade a *posteriori* de observar y_i quando o modelo é ajustado sem y_i . Quanto maior o valor da estatística CPO melhor é o ajuste do modelo aos dados. Ao invés de ajustar o modelo a cada vez que é retirada uma observação, uma aproximação do CPO pode ser obtida por amostragem da distribuição a *posteriori*:

$$CPO_i = \left(\frac{1}{T \sum_{t=1}^T p(\mathbf{y}_i|\boldsymbol{\theta}^t)} \right)^{-1},$$

em que $\boldsymbol{\theta}^t$ é a amostra de $\prod(\boldsymbol{\theta}|\mathbf{y}_i)$ e T indica o número de amostras a *posteriori*.

Ross e Held (2011) indicam o uso da versão “logaritimizada”, definida por $\log CPO = -(n-1) \sum_{i=1}^n \log p(\mathbf{y}_i|\mathbf{y}_{-i})$. Assim, valores mais baixos tanto para DIC quanto para logCPO são indicadores de melhor ajuste e qualidade preditiva, respectivamente.

Geralmente, os três critérios não concordam conjuntamente o mesmo modelo, assim, considerou-se primeiro os modelos que apresentaram melhor desempenho baseado no critério logCPO. Nos casos em que os valores do logCPO foram similares entre dois ou mais modelos usou-se o WAIC e como último critério para escolha do modelo o critério, usou-se DIC.

2.4 Estudo de simulação

Nesta seção, um estudo de simulação de Monte Carlo foi realizado para avaliar o impacto da escolha de bordas não convexas no comportamento das estimativas obtidas por meio da abordagem INLA.

Para comparar as diferentes bordas não convexas foram realizadas 200 simulações de Monte Carlo no *software R*. Para cada réplica, uma amostra geoestatística espacial com tamanho amostral n foi obtida. Neste estudo de simulação, considerou-se tamanhos amostrais iguais a $n = 50$, $n = 100$ e $n = 200$. Os locais observados foram gerados aleatoriamente em um quadrado unitário.

Os dados foram gerados de acordo com o modelo $y = \beta_0 + s + e$, em que $\mathbf{s}$ é um campo aleatório espacial de Matérn com função de correlação dada por $Cor_M(X(\mathbf{s}_i), X(\mathbf{s}_j)) = \frac{1}{2^{\nu-1}\gamma(\nu)} (\kappa \|\mathbf{s}_i - \mathbf{s}_j\|)^\nu K_\nu(\kappa \|\mathbf{s}_i - \mathbf{s}_j\|)$, e e é o ruído branco. Os valores escolhidos para a variabilidade espacial do campo aleatório espacial Matérn e para o ruído branco foram, $\sigma_s^2 = 5$ e $\sigma_e^2 = 2$, respectivamente. O alcance nominal para o campo aleatório espacial é especificado por $\kappa^* = \sqrt{8\nu}/\kappa = 0,4$, definido pelo parâmetro de suavização $\kappa = 7$ e o parâmetro de escala $\nu = 1$. A média geral é dada por $\beta_0 = 10$.

Para a construção da malha foi utilizada a função *inla.mesh.2d*. Um dos argumentos dessa função é o *max.edge(s1,s2)*, que define comprimento máximo das arestas no domínio interno e externo. Os parâmetros *s1* e *s2* foram fixados em 0,06 e 0,2, respectivamente. Para investigar o impacto da borda não convexa, diferentes escolhas de bordas foram consideradas no estudo. Diferentes bordas são consideradas alterando os parâmetros da função *inla.nonconvex.hull* definidas pelos argumentos *convex* (*a*) e *resolution* (*b*). O conjunto de valores para o raio de extensão desejado (*a*) foram baseados nas distâncias entre as observações. Os valores utilizados correspondem a 7, 15, 25 e 35 por cento da maior distância entre as observações. Para o parâmetro correspondente a resolução interna do cálculo foi utilizado o conjunto de valores $b=(25,40,60)$. Combinando estes dois argumentos tem-se um total de 12 bordas não convexas diferentes.

A Tabela 2.1 contém, para todas as combinação diferentes dos parâmetros *a* e *b*, o número total de vértices em cada malha. Uma triangulação mais densa é obtida com os maiores valores de *a* e *b*.

Tabela 2.1: Número nós para diferentes combinações dos parâmetros *a* e *b* para os tamanho da amostra.

<i>b</i>	n=50				n=100				n=200			
	7%	15%	25%	35%	7%	15%	25%	35%	7%	15%	25%	35%
25	844	1256	1780	2379	959	1377	1893	2511	1038	1453	2008	2667
40	856	1273	1802	2374	987	1398	1899	2510	1043	1433	2017	2636
60	929	1294	1785	2396	1020	1397	1936	2489	1075	1475	1991	2615

Para investigar o efeito das bordas não convexas na inferência, foi calculada, para

cada malha, a probabilidade de cobertura (baseada no intervalo de 95% de confiança), o viés e o erro quadrático médio (EQM) para cada um dos parâmetros do modelo.

Para verificação das probabilidades de cobertura foram construídos intervalos de confiança para os parâmetros. Ainda, calculou-se quantos intervalos continham os verdadeiros valores dos parâmetros para todas as combinações especificadas. As porcentagens de cobertura são dadas por: $((200 - \text{n}^\circ \text{ de intervalos que não contem o verdadeiro valor do parâmetro})/200) \times 100\%$. Na Tabela 2.2, são apresentadas as probabilidades de cobertura para os parâmetros da função de covariância (σ_s^2 , σ_e^2 e κ^*) e para a média geral β_0 .

Tabela 2.2: Porcentagens de cobertura dos intervalos de credibilidade 95% para os parâmetros estimados para cada distribuição.

n	a	b	Parâmetros			
			β_0	σ_s^2	σ_e^2	κ^*
50	7%	25	82,5%	17,0%	81,5%	27,5%
		40	88,0%	25,0%	80,5%	34,5%
		60	83,0%	16,0%	80,0%	26,5%
	15%	25	85,5%	27,5%	86,5%	36,5%
		40	88,5%	28,5%	87,5%	38,0%
		60	87,0%	27,0%	86,0%	37,0%
	25%	25	87,5%	35,0%	87,5%	43,0%
		40	86,5%	36,0%	87,0%	44,0%
		60	90,0%	43,0%	90,0%	49,5%
	35%	25	90,5%	46,0%	89,0%	53,0%
		40	91,5%	44,0%	89,0%	52,5%
		60	91,5%	47,5%	89,5%	54,0%
100	7%	25	93,5%	92,0%	95,5%	94,0%
		40	93,0%	92,0%	95,5%	95,0%
		60	93,0%	92,0%	95,5%	93,5%
	15%	25	93,5%	91,0%	96,5%	94,0%
		40	93,5%	91,5%	96,0%	94,0%
		60	93,5%	90,5%	95,5%	93,5%
	25%	25	93,5%	91,0%	95,0%	92,5%
		40	93,5%	91,0%	95,5%	93,0%
		60	93,5%	91,0%	95,5%	92,0%
	35%	25	93,5%	91,0%	95,5%	89,5%
		40	94,0%	91,0%	96,0%	90,0%
		60	94,0%	91,0%	95,5%	90,5%
200	7%	25	96,0%	93,0%	95,5%	91,0%
		40	96,0%	93,5%	95,5%	91,0%
		60	96,0%	93,0%	95,5%	90,5%
	15%	25	96,0%	93,0%	94,0%	90,5%
		40	96,0%	93,0%	94,0%	90,5%
		60	96,0%	93,5%	94,5%	90,5%
	25%	25	96,0%	93,5%	91,5%	87,0%
		40	96,0%	93,5%	92,5%	86,5%
		60	96,0%	93,5%	93,0%	86,5%
	35%	25	96,0%	92,5%	91,5%	85,5%
		40	96,0%	93,0%	91,5%	85,5%
		60	96,0%	93,0%	91,0%	85,5%

Pode-se observar que para amostras de tamanho $n = 50$ o maior impacto na

probabilidade de cobertura é observado para o parâmetro *convex*. Os parâmetros β_0 e σ_e^2 apresentaram boa cobertura, observada principalmente para bordas com altos valores para *convex*. Para os valores do argumento *resolution* pode-se observar um pequeno efeito sobre a probabilidade de cobertura, apresentando maior impacto no parâmetro σ_s^2 . A malha com *convex* = 35% e *resolution* = 60, apresenta melhor cobertura para todos os parâmetros considerando amostras de tamanho $n = 50$. Para amostras de tamanho $n = 100$ e $n = 200$, os parâmetros *convex* e *resolution* não apresentaram efeito sobre a probabilidade de cobertura em todos os parâmetros analisados. No entanto, observou-se que a taxa de cobertura para todos os parâmetros foi superior a 85%.

Tabela 2.3: Erro quadrático médio (EQM) para os parâmetros estimados para cada malha.

n	a	b	Parâmetros			
			β_0	σ_s^2	σ_e^2	κ^*
50	7%	25	0,8396	3,3724	79,0112	0,0814
		40	0,8390	3,0495	9,9397	0,1199
		60	0,8421	3,4100	777,2401	0,0828
	15%	25	0,8570	3,0192	25,6721	0,1650
		40	0,8454	2,9682	10,2322	0,1663
		60	0,8564	2,9790	53,8003	0,1684
	25%	25	0,8872	2,6959	18,0773	0,2187
		40	0,8745	2,6475	37,7282	0,1636
		60	0,8570	2,3626	18,2569	0,1732
	35%	25	0,8604	2,2003	26,9052	0,3803
		40	0,8898	2,2622	28,1735	0,4816
		60	0,8714	2,1691	34,5606	0,4420
100	7%	25	0,7289	0,3068	5,1516	0,0498
		40	0,7278	0,3147	5,1148	0,0491
		60	0,7297	0,3177	5,1418	0,0491
	15%	25	0,7610	0,3096	7,4680	0,0702
		40	0,7630	0,3108	7,5833	0,0714
		60	0,7638	0,3109	7,5253	0,0710
	25%	25	0,7926	0,3089	10,5312	0,0948
		40	0,7944	0,3086	10,5569	0,0954
		60	0,7932	0,3072	10,4588	0,0935
	35%	25	0,8126	0,3106	13,7460	0,1152
		40	0,8120	0,3103	13,7349	0,1144
		60	0,8131	0,3101	13,6882	0,1128
200	7%	25	0,6735	0,0998	10,4787	0,0543
		40	0,6732	0,0997	10,4125	0,0546
		60	0,6734	0,0998	10,4728	0,0545
	15%	25	0,7090	0,1002	16,4470	0,0721
		40	0,7100	0,0997	16,7961	0,0727
		60	0,7098	0,1002	16,7409	0,0726
	25%	25	0,7417	0,1004	24,9476	0,0940
		40	0,7410	0,1006	24,7870	0,0935
		60	0,7408	0,1010	24,3204	0,0936
	35%	25	0,7626	0,0997	32,4258	0,1129
		40	0,7630	0,1008	32,7298	0,1141
		60	0,7612	0,1004	31,8135	0,1119

A Tabela 2.3 apresenta o erro quadrático médio dos parâmetros estimados, para todas as malhas. Observa-se que o parâmetro σ_e^2 foi o que sofreu maior impacto com a mudança dos parâmetros *convex* e *resolution*. Note que para amostras de tamanho $n = 100$ e $n = 200$ os maiores valores do EQM são observados para valores altos de *convex*. Para as amostras de tamanho $n = 50$ o maior impacto é observado para o maior valor do parâmetro *resolution*. Para o parâmetro β_0 a melhor combinação, para todos os tamanhos amostrais analisados, foi *convex* = 7% e *resolution* = 40. Para parâmetro σ_s^2 em

amostras de tamanho $n = 50$, a melhor malha é obtida para $convex = 35\%$ e $resolution = 60$. Como a variação entre os EQM do parâmetro σ_s^2 foi muito pequena, escolhemos a malha com os menores valores de $convex$ e $resolution$ por apresentarem o menor número de vértices, fazendo com que o custo computacional seja o menor possível. Para o parâmetro σ_e^2 , as melhores malhas são obtidas com os valores de $convex$ e $resolution$ iguais 7% e 40, respectivamente. Observando todos os parâmetros, podemos concluir que a melhor com os menores valores $convex$ e $resolution$ produziram resultados ótimos nessa configuração de simulação.

Os resultados apresentados anteriormente mostraram que há um grande impacto da malha na inferência e que não existe uma malha única que seja ótima. Agora, investigamos se podemos usar os critérios de informação preditiva, conforme descrito na Seção 2.3, para selecionar uma malha ótima.

Para realizar a seleção de malha, calculamos os valores médios do DIC, WAIC, logCPO e tempo computacional para cada simulação executada. Estes valores são apresentados na Tabela 2.4. Podemos observar que para as amostras de tamanho $n = 100$ e $n = 200$, existem várias malhas com bons resultados, e uma forma de escolher uma delas é por meio do número mínimo de vértices, que ocorre para os menores valores de $convex$ e $resolution$ (Tabela 2.1). Considerando estes resultados podemos escolher para amostra de tamanho $n = 100$ e $n = 200$, as malhas com $convex = 7\%$ e $resolution = 60$ e $convex = 7\%$ e $resolution = 25$, respectivamente. Combinando os resultados apresentados na Tabela 2.4 e Tabela 2.3, podemos concluir que para obtermos os melhores resultados, devemos utilizar as malhas confeccionadas com os menores valores de $convex$ e $resolution$. Para as amostras de tamanho $n = 50$ os menores valores dos critérios DIC e WAIC foram observados para os maiores valores de $convex$. Baseado nos resultados apresentados nas Tabelas 2.4 e 2.3, podemos concluir que para esse tamanho de amostra, as malhas que apresentam os melhores resultados são as malhas cujos valores dos parâmetros $convex$ e $resolution$ são 25% e 60, respectivamente.

Tabela 2.4: Valores médios do DIC, logCPO, WAIC e tempo computacional para diferentes distribuições.

n	a	b	DIC	logCPO	WAIC	Tempo
50	7%	25	272,0985	2,084479	278,2139	63,7410
		40	258,6291	2,080748	264,2276	59,2046
		60	272,9214	2,083964	278,9249	64,2755
	15%	25	258,8153	2,094060	265,6551	81,6670
		40	257,6208	2,095799	263,9410	77,0325
		60	257,9959	2,091449	264,5280	76,7411
	25%	25	248,1965	2,081762	254,6432	98,9031
		40	250,0967	2,067659	252,7587	102,1006
		60	235,3854	2,085319	239,4576	97,0371
	35%	25	221,5970	2,086990	228,1150	99,4445
		40	224,2790	2,075224	228,6073	101,4235
		60	222,3710	2,083313	227,8828	103,4562
100	7%	25	382,0837	2,010075	382,7591	25,4963
		40	381,8292	2,010147	382,5803	25,7991
		60	381,4852	2,009764	382,2708	26,3731
	15%	25	382,3813	2,009826	383,1332	36,4095
		40	382,2353	2,009648	382,9360	37,4199
		60	382,2185	2,009682	382,9214	38,9045
	25%	25	382,3935	2,009579	383,1556	51,7097
		40	382,5614	2,009732	383,2961	52,8212
		60	382,5079	2,009728	383,2707	53,2592
	35%	25	382,4959	2,009417	383,2225	75,8046
		40	382,5275	2,009448	383,2668	73,3468
		60	382,5181	2,009445	383,2435	75,6745
200	7%	25	757,7149	1,935985	759,3351	35,0816
		40	757,7881	1,935927	759,4042	36,5714
		60	757,7275	1,935957	759,3443	37,1478
	15%	25	757,8572	1,935862	759,5445	50,7814
		40	757,8375	1,935853	759,5505	49,8704
		60	757,8461	1,935811	759,5414	49,9233
	25%	25	757,7700	1,935758	759,4939	71,7983
		40	757,7899	1,935724	759,4905	71,2574
		60	757,7561	1,935677	759,4425	70,2236
	35%	25	757,8974	1,935686	759,6307	108,8899
		40	757,7988	1,935683	759,4979	100,5857
		60	757,8209	1,935600	759,5430	103,8315

2.5 Conclusão

Ao usar a abordagem SPDE no INLA na análise geoestatística, o primeiro passo é criar uma malha sobre a área de estudo. A pré-especificação dessa malha não é uma tarefa simples. Com um estudo de simulação, mostramos que os resultados inferenciais dependem dessa malha criada. Os resultados da simulação indicaram que o valor de a , que determina o menor raio de curvatura convexa permitida, tem um impacto significativo nos resultados. Por outro lado, a escolha do valor de b , apresenta um pequeno impacto nos resultados nas amostras de tamanho $n = 100$ e $n = 200$, e um grande impacto nas amostras de tamanho $n = 50$.

Referências

BOLIN, D.; LINDGREN, F. A comparison between Markov approximations and other methods for large spatial data sets. *Computational Statistics and Data Analysis*, Springer, v. 61, p. 7–21, 2013.

GEISSER, S.; EDDY, W. A predictive approach to model selection. *Journal of the American Statistical Association*, Boston, v. 74, n. 365, p. 153–160, 1979.

GELFAND, A. E.; DIGGLE, P. J.; FUENTES, M.; GUTTORP, P. *Handbook of spatial statistics*. CRC press, 2010.

GELMAN, A.; HWANG, J.; VEHTARI, A. Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, Springer, v. 24, n. 6, p. 997–1016, 2013.

GHOSH, P.; HUANG, L.; YU, B.; TIWARI, R. C. Semiparametric Bayesian approaches to joinpoint regression for population-based cancer survival data. *Computational statistics and data analysis*, Elsevier, v. 53, n. 12, p. 4073–4082, 2009.

LINDGREN, F.; RUE, H.; LINDSTRÔM, J. An explicit link between Gaussian fields and Gaussian Markov random fields: the SPDE approach (with discussion). *J. R. Stat. Soc. Ser. B.*, v. 73, p. 423–498.

LINDGREN, F. Continuous domain spatial models in R-INLA. *The ISBA Bulletin*, v. 19, n.4, 2012.

LINDGREN, F. Bayesian spatial modelling with R-INLA. *J. Stat. Software*, v. 63, 2015.

BOLIN, D.; LINDGREN, F. The Matérn function as a general model for soil variograms. *Geoderma*, Elsevier, v. 128, p. 192–207, 2005.

RIGHETTO, A. J. *The impact of choosing different meshes under INLA/SPDE framework for geostatistical modelling*. 2017. 95p. Tese (Doutorado em Estatística e Experimentação Agronômica) - Escola Superior de Agricultura “Luiz de Queiroz”, Universidade de São Paulo, Piracicaba, 2017.

ROSS, M.; HELD, L. Sensitivity analysis in Bayesian generalized linear mixed models for binary data. *Bayesian Analysis*, v. 6, n. 2, p. 259–278, 2011.

RUE, H.; MARTINO, S.; CHOPIN, N. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations (with discussion). *Journal of the Royal Statistical Society–Series B*, Wiley Online Library, v. 71, Part 2, p. 319–392, 2009.

SPIEGELHALTER, D. J.; BEST, N. G.; CARLIN, B. P.; LINDE, A. V. D. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society–Series B*, Wiley Online Library, v. 64, n. 4, p. 583–639, 2000.

WATANABE, S. Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, v. 11, n. 2, p. 3571–3594, 2010.

3 IMPACTO DA ESCOLHA DO MÉTODO DE APROXIMAÇÃO PARA A DISTRIBUIÇÃO CONDICIONAL DO CAMPO ALEATÓRIO NOS MODELOS LINEARES GENERALIZADOS

Resumo: A família dos modelos gaussianos latentes é adaptável a uma grande quantidade de aplicações que requerem modelagem complexa, inclusive os modelos lineares generalizados. A inferência nesta classe de modelos é normalmente realizada usando métodos computacionalmente intensivos, tais como, os algoritmos Monte Carlo via Cadeia de Markov (MCMC). Apesar de serem métodos altamente eficientes, a convergência do algoritmo em modelos complexos pode ser extremamente lenta, o que gera alto custo computacional. Em 2009 surgiu o método de Aproximações de Laplace Aninhadas Integradas (*Integrated Nested Laplace Approximation* - INLA) que busca eficiência tanto no custo computacional como na precisão das estimativas. O pacote R-INLA consta com três métodos para aproximar a posteriori conjunta $\pi(\boldsymbol{\theta}|\mathbf{y})$: aproximação gaussiana, aproximação de Laplace e aproximação simplificada de Laplace. Neste trabalho, investigamos o impacto da escolha dos métodos de aproximação em modelos lineares generalizados por meio de estudos de simulação. As distribuições utilizadas para gerar os dados analisados neste trabalho foram a Binomial, Poisson, Gama e Beta. Em todos os modelos, medidas de concordância entre as observações e os modelos foram tomadas, foram elas: erro quadrático médio e taxa de cobertura.

Abstract: The family of latent Gaussian models is adaptable to a large number of applications that require complex modeling, including generalized linear models. Inference in this class of models is usually accomplished using computationally intensive methods, such as Markov Chain Monte Carlo (MCMC) algorithms. Although they are highly efficient methods, the convergence of the algorithm in complex models can be extremely slow, which generates high computational cost. In 2009 came the Integrated Nested Laplace Approximation (INLA) method that seeks efficiency both in the computational cost and in the precision of the estimates. The R-INLA package has three methods to approximate the posteriori $\pi(\boldsymbol{\theta}|\mathbf{y})$: Gaussian approximation, Laplace approximation and simplified Laplace approximation. In this work, we investigate the impact of the choice of approximation methods in generalized linear models through simulation studies. The distributions used to generate the data analyzed in this work were Binomial, Poisson, Gamma and Beta. In all the models, agreement measures between the observations and the models were taken, they were: average quadratic error and coverage rate.

Keywords: Linear generalized model; Laplace approximation; Hierarchical models; Bayesian inference

3.1 Introdução

Em experimentos científicos vários tipos de respostas podem ser observadas, dentre elas respostas do tipo discreta e contínua. Nestes casos a resposta deixa de ter, obrigatoriamente, distribuição normal, podendo ser modelada por outra distribuição de probabilidade. Esta questão foi considerada e resolvida com a introdução dos modelos lineares generalizados, propostos por Nelder e Wedderburn (1972), permitindo que fenômenos aleatórios fossem modelados de acordo com distribuição apropriada sem a necessidade de realizar transformações nos dados observados.

Como apontado anteriormente, com a introdução dos modelos lineares generalizados (MLG's), a variável resposta deixa de ter, obrigatoriamente, distribuição normal, passando a ter distribuição pertencente a uma família de distribuições chamada de família exponencial. Uma característica importante dessa família é que a relação entre o valor médio da variável resposta e a combinação linear das variáveis explanatórias pode ser estabelecida por qualquer função monótona e diferenciável, o que antes era obtido apenas por meio da função identidade (RIZZATO, 2011). Muitas distribuições podem ser acomodadas na família exponencial, incluindo, por exemplo, as distribuições Binomial e Poisson para dados de contagem, a distribuição Gama para dados contínuos positivos e a distribuição Beta para modelar proporção ou dados pertencentes ao intervalo (0,1).

Segundo Rocha (2015), nos modelos lineares generalizados, o ajuste baseado em verossimilhança pode apresentar problemas, o que se torna mais notório no caso de pequenas amostras, haja visto que a inferência se baseia na distribuição assintótica dos estimadores de máxima verossimilhança. Como solução para este fato, propõe-se a abordagem bayesiana. Os métodos mais difundidos na estimação bayesiana para obtenção das distribuições conjuntas a *posteriori* são os métodos de Monte Carlo com Cadeias de Markov (MCMC). Estes métodos exigem que as simulações converjam para que tenhamos certeza de que temos uma distribuição a *posteriori* confiável. Nos últimos anos, um novo método foi proposto por Rue, Martino e Chopin (2009), denominado INLA (*Integrated Nested Laplace Approximation*), que é uma alternativa computacionalmente eficaz aos métodos baseados em algoritmos de MCMC.

A metodologia INLA foi proposta para realizar inferência bayesiana aproximada na classe dos modelos Gaussianos latentes (MGL), uma subclasse de modelos de regressão aditiva estruturada. Esses modelos são formulados de forma muito similar aos modelos lineares generalizados, mas o preditor linear é substituído por um preditor aditivo estruturado que pode incluir efeitos não-lineares. O INLA faz uso de aproximações determinísticas de Laplace aninhadas e, como um algoritmo feito sob medida para a classe de LGMs, fornece uma alternativa mais rápida e precisa, principalmente por fazer uso de matrizes esparsas, que são inerentes a essa classe de modelos, superando o método MCMC em termos de acurácia e tempo computacional (Martins *et al.*, 2013).

O INLA consiste em construir aproximações de Laplace, em que originalmente se aproximava as distribuições a *posteriori*. O primeiro passo na aproximação INLA é realizar uma aproximação de Laplace para a *posteriori* conjunta

$$\pi(\boldsymbol{\theta}|\mathbf{y}) = \frac{\pi(\boldsymbol{\theta})\pi(\mathbf{x},\boldsymbol{\theta})\pi(\mathbf{y}|\mathbf{x})}{\pi(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})} \propto \frac{\pi(\boldsymbol{\theta})\pi(\mathbf{x},\boldsymbol{\theta})\pi(\mathbf{y}|\mathbf{x})}{\pi_G(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})},$$

em que $\pi_G(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$ é a aproximação Gaussiana para $\pi(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$, que corresponde à verdadeira distribuição da moda (Rue, Martino e Chopin, 2009). As aproximações para as distribuições a *posteriori* marginais, para os parâmetros não-Gaussianos, podem então ser construídas por meio de integração numérica, desde que a dimensão de $\boldsymbol{\theta}$ não seja muito grande. As distribuições a *posteriori* marginais para o campo latente $\pi(\mathbf{x}_i|\mathbf{y})$ são construídas como uma aproximação de Laplace para $\pi(\mathbf{x}_i|\boldsymbol{\theta},\mathbf{y})$ e, em seguida, integrando-se em relação à distribuição a *posteriori* conjunta aproximada para $\boldsymbol{\theta}|\mathbf{y}$. Todos os detalhes do esquema de aproximação podem ser encontrados em Rue, Martino e Chopin (2009). Outra forma de aproximar as distribuições a *posteriori* marginais para o campo latente é a utilização da forma simplificada de Laplace. Ainda de acordo com os autores, a diferença entre os três métodos de aproximação variam em precisão e custo computacional.

Desta maneira, o objetivo do presente trabalho é estudar exaustivamente os métodos de aproximação para as marginais da distribuição condicional do campo latente, ver seu impacto na análise e definir diretrizes para utilização dos mesmo em análises em que a variável resposta pode ser modelada por meio de uma das distribuições pertencentes à família exponencial, especificamente, as distribuições Binomial, Poisson, Beta e Gama.

3.2 INLA

Na inferência bayesiana, o maior interesse está na obtenção das distribuições a *posteriori* marginais para os parâmetros desconhecidos do modelo. A obtenção de tais distribuições para modelos complexos, implica em um alto custo computacional ou até mesmo na falta de convergência. Como alternativa, Rue, Martino e Chopin (2009) propuseram o método INLA.

O método INLA permite a aproximação das distribuições marginais a *posteriori* dos modelos de regressão com estrutura aditiva, denominados de modelos gaussianos latentes. Nesses modelos, considera-se que a distribuição da variável resposta Y_i pertence a uma determinada família de distribuições (não necessariamente à família exponencial) e que estamos interessados em um parâmetro μ_i que representa a média desta distribuição. Considera-se ainda que a média μ_i está ligada a uma estrutura aditiva do preditor linear η_i por meio de uma função de ligação $g(\cdot)$, tal que $g(\mu_i) = \eta_i$, em que η_i pode ser apresentado

da seguinte forma,

$$\boldsymbol{\eta}_i = \boldsymbol{\alpha} + \sum_{j=1}^{n_f} f^{(j)}(\mathbf{u}_{ji}) + \sum_{k=1}^{n_\beta} \beta_k \mathbf{z}_{ki} + \boldsymbol{\varepsilon}_i,$$

em que β_k representa o efeito linear de covariáveis $\mathbf{z}$, $f^{(h)}(\cdot)$ são funções não conhecidas de covariáveis $\mathbf{u}$ e $\boldsymbol{\varepsilon}_i$ são os termos não estruturados, utilizados para acomodar a correlação espacial, temporal e/ou espaço-temporal, caso exista.

Segundo Rue, Martino e Chopin (2009), a definição dos modelos gaussianos latentes (MGL) é dada como sendo modelos que assumem a forma aditiva apresentada em 3.1 e assume-se *priori* gaussiana para todos os parâmetros que compõe $\boldsymbol{\eta}$, o vetor desses parâmetros é chamado de campo aleatório gaussiano (CAG). Neste trabalho será considerada apenas aplicações em modelos lineares generalizados. Segundo Dey, Ghosh e Mallick (2000), o preditor linear dos modelos lineares generalizados pode ser expresso por,

$$\eta_i = \alpha + \sum_{k=1}^{n_\beta} \beta_k \mathbf{z}_{ki}. \quad (3.1)$$

Segundo Martins *et al.* (2013), o modelo descrito em 3.1 pode ser escrito utilizando uma estrutura hierárquica, em que o primeiro estágio é formado pela função de verossimilhança com propriedades de independência condicional, dada pelo campo latente gaussiano $\mathbf{x} = (\boldsymbol{\eta}, \alpha, \boldsymbol{\beta})$ e possíveis hiperparâmetros $\boldsymbol{\theta}$ (associados aos componentes de variância). Assim, o primeiro estágio da hierarquia é dado por,

$$\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_1 \sim \pi(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_1) = L(\mathbf{x}, \boldsymbol{\theta}_1|\mathbf{y}) = \prod_{i=1}^{n_d} \pi(y_i|x_i, \boldsymbol{\theta}_1).$$

A distribuição condicional do campo latente $\mathbf{x}$, dados alguns possíveis hiperparâmetros $\boldsymbol{\theta}_2$ forma o segundo estágio do modelo e tem uma distribuição Gaussiana conjunta.

$$\mathbf{x}|\boldsymbol{\theta}_2, \mathbf{y} \sim \pi(\mathbf{x}|\boldsymbol{\theta}_2, \mathbf{y}) = N(\boldsymbol{\mu}(\boldsymbol{\theta}_2), \mathbf{Q}^{-1}(\boldsymbol{\theta}_2)),$$

em que $N(\boldsymbol{\mu}, \mathbf{Q}^{-1})$ denota a distribuição gaussiana multivariada com vetor de média $\boldsymbol{\mu}$ e matriz de precisão $\mathbf{Q}$. Na maioria das aplicações, os campos Gaussianos latentes possuem propriedades de independência condicional, o que se traduz em uma matriz de precisão esparsa $\mathbf{Q}(\boldsymbol{\theta}_2)$, que é de extrema importância para reduzir o custo computacional. Uma distribuição Gaussiana multivariada com uma matriz de precisão esparsa é conhecida como um campo aleatório de Markov de Gauss (GMRF) (RUE; HELD, 2005).

O terceiro e último estágio do modelo hierárquico é dado pela especificação de uma distribuição *a priori* apropriada para cada um dos hiperparâmetros do modelo,

$$\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta}).$$

Para o modelo descrito anteriormente, a distribuição conjunta a *posteriori* para os parâmetros do modelo linear generalizado, pode ser expressa por,

$$\begin{aligned}\pi(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y}) &\propto \pi(\boldsymbol{\theta}) \pi(\mathbf{x} | \boldsymbol{\theta}) \prod_{i=1}^{n_d} \pi(y_i | x_i, \boldsymbol{\theta}) \\ &\propto \pi(\boldsymbol{\theta}) |\mathbf{Q}(\boldsymbol{\theta})|^{n/2} \exp \left[-\frac{1}{2} \mathbf{x}^t \mathbf{Q}(\boldsymbol{\theta}) \mathbf{x} + \sum_{i=1}^{n_d} \log \pi(y_i | x_i, \boldsymbol{\theta}) \right].\end{aligned}\quad (3.2)$$

Segundo Rue, Martino e Chopin (2009), as distribuições marginais a *posteriori* de interesse podem ser expressas por,

$$\begin{aligned}\pi(x_i | \mathbf{y}) &= \int \pi(x_i | \boldsymbol{\theta}, \mathbf{y}) \pi(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}, \\ \pi(\theta_j | \mathbf{y}) &= \int \pi(\boldsymbol{\theta} | \mathbf{y}) \boldsymbol{\theta}_{-j},\end{aligned}$$

em que $i = 1, \dots, I$, sendo I o número de parâmetros de efeito fixo e efeito aleatório do modelo, e $j = 1, \dots, J$, sendo J o número de componentes de variância. Rue, Martino e Chopin (2009) utilizam as equações anteriores para construir as aproximações aninhadas,

$$\tilde{\pi}(x_i | \mathbf{y}) = \int \tilde{\pi}(x_i | \boldsymbol{\theta}, \mathbf{y}) \tilde{\pi}(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}, \quad (3.3)$$

$$\tilde{\pi}(\theta_j | \mathbf{y}) = \int \tilde{\pi}(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}_{-j}. \quad (3.4)$$

Para as equações (3.3) e (3.4), Rue, Martino e Chopin (2009) sugerem dividir o método de aproximação INLA em três etapas; Primeiramente propor uma aproximação $\tilde{\pi}(\boldsymbol{\theta} | \mathbf{y})$ para a distribuição conjunta a *posteriori* dos hiperparâmetros $\pi(\boldsymbol{\theta} | \mathbf{y})$. O segundo passo consiste em propor uma aproximação $\tilde{\pi}(x_i | \boldsymbol{\theta}, \mathbf{y})$ para as marginais da distribuição condicional do campo latente dado as observações e os hiperparâmetros $\pi(x_i | \boldsymbol{\theta}, \mathbf{y})$, e finalmente explorar $\tilde{\pi}(\boldsymbol{\theta} | \mathbf{y})$ em uma grade de valores e usá-los para integrar $\boldsymbol{\theta}$ na Equação (3.3) e $\boldsymbol{\theta}_{-j}$ na Equação (3.4).

3.2.1 Aproximação para $\pi(\boldsymbol{\theta} | \mathbf{y})$

A aproximação utilizada para a obtenção da distribuição marginal a *posteriori* dos hiperparâmetros $\pi(\boldsymbol{\theta} | \mathbf{y})$, segundo Martins *et al.* (2013), é expressa por,

$$\tilde{\pi}(\boldsymbol{\theta} | \mathbf{y}) \propto \frac{\pi(\mathbf{x}, \boldsymbol{\theta}, \mathbf{y})}{\pi_G(\mathbf{x} | \boldsymbol{\theta}, \mathbf{y})} \Big|_{\mathbf{x}=\mathbf{x}^*(\boldsymbol{\theta})}, \quad (3.5)$$

em que $\tilde{\pi}_G(\mathbf{x} | \boldsymbol{\theta}, \mathbf{y})$ é a aproximação gaussiana para a condicional completa de $\mathbf{x}$, obtida combinando a configuração modal e a curvatura na moda e $\mathbf{x}^*(\boldsymbol{\theta})$ é a moda da condicional

completa de $\mathbf{x}$, para um dado $\boldsymbol{\theta}$. Segundo Martins *et al.* (2013), a aproximação definida em (3.5) é exata se $\pi(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ é gaussiana.

3.2.2 Aproximação para $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$

Segundo Rue, Martino e Chopin (2009), para aproximar a distribuição marginal a *posteriori* de cada um dos elementos de $\mathbf{x}$, três métodos podem ser considerados: aproximação gaussiana, aproximação de Laplace e aproximação simplificada de Laplace.

A diferença entre os três métodos é baseada na precisão e tempo computacional. A opção mais rápida computacionalmente para $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$ é a aproximação Gaussiana $\tilde{\pi}_G(x_i|\boldsymbol{\theta}, \mathbf{y})$, uma vez que já foi calculada durante a exploração de $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$, restando apenas as variâncias marginais serem calculadas. Segundo Rue e Martino (2007), é a opção mais rápida de ser obtida e oferece resultados razoáveis, porém pode apresentar erros de locação e/ou erros devido à falta de assimetria da aproximação.

A abordagem mais precisa é a aproximação de Laplace, denotada por $\tilde{\pi}_{LA}(x_i|\boldsymbol{\theta}, \mathbf{y})$, expressa por,

$$\tilde{\pi}_{LA}(x_i|\boldsymbol{\theta}, \mathbf{y}) \propto \frac{\pi(\mathbf{x}, \boldsymbol{\theta}, \mathbf{y})}{\tilde{\pi}_{GG}(\mathbf{x}_{-i}|x_i, \boldsymbol{\theta}, \mathbf{y})} \Big|_{\mathbf{x}_{-i}=\mathbf{x}_{-i}^*(x_i, \boldsymbol{\theta})} \quad (3.6)$$

em que $\mathbf{x}_{-i}$ representa o vetor $\mathbf{x}$ com o i -ésimo elemento retirado, $\tilde{\pi}_{GG}(\mathbf{x}_{-i}|x_i, \boldsymbol{\theta}, \mathbf{y})$ é a aproximação Gaussiana para $\mathbf{x}_{-i}|x_i, \boldsymbol{\theta}, \mathbf{y}$ e $\mathbf{x}_{-i}^*$ é a configuração modal. Observe que o denominador em (3.6) é diferente do denominador definido em (3.5), que implica no cálculo de $\tilde{\pi}_{GG}$ para cada valor de x_i e $\boldsymbol{\theta}$, o que torna o custo computacional alto. Ou seja, a aproximação de Laplace oferece resultados mais precisos e é exata quando a condicional completa $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$ for Gaussiana (o que não ocorre quando a verossimilhança não é gaussiana), entretanto, tem um alto custo computacional.

A aproximação de Laplace simplificada para $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$, denotada por $\tilde{\pi}_{SLA}(x_i|\boldsymbol{\theta}, \mathbf{y})$, baseia-se na expansão em série de Taylor no numerador e denominador da Equação (3.6) em torno de $x_i = \mu_i(\boldsymbol{\theta})$ e mantendo os termos de terceira ordem. De acordo com Rue e Martino (2007), essa medida permite corrigir a aproximação gaussiana $\tilde{\pi}_G(x_i|\boldsymbol{\theta}, \mathbf{y})$ para localização e assimetria. O benefício é puramente computacional.

3.3 Seleção de modelos e medidas para avaliação do ajuste

Para avaliar e comparar o desempenho dos modelos, diferentes critérios de informação preditiva podem ser usados. Conforme em Ghosh *et al.* (2009) três abordagens para seleção Bayesiana de modelos são considerados neste estudo. A primeira é o Critério de Informação Deviance (*Deviance Information Criterion* - DIC) (Spiegelhalter *et al.* (2000)), o critério de informação de Watanabe-Akaike (WAIC) e as condicionais preditivas ordenadas (*Conditional predictive ordinate* - CPO).

Para o cálculo do DIC considera-se a medida *deviance* dada por $D = -2 \log p(\mathbf{y}|\hat{\boldsymbol{\theta}})$, em que $\mathbf{y}$ é o vetor de dados, $\boldsymbol{\theta}$ é o vetor de parâmetros de interesse sob determinado modelo e $p(\mathbf{y}|\boldsymbol{\theta})$ a função de verossimilhança. O DIC é calculado como $DIC = \hat{D} + 2p_D$, em que $\hat{D} = E_{\boldsymbol{\theta}|\mathbf{y}}[D]$ é uma medida que indica a qualidade do ajuste, e p_D o número efetivo de parâmetros, representando a complexidade do modelo. Segundo Spiegelhalter *et al.* (2000), a quantidade efetiva de parâmetros do modelo é calculado como a média a *posteriori* da *deviance* menos a *deviance* avaliada na média a *posteriori*, ou seja, $p_D = \hat{D} - D(\hat{\boldsymbol{\theta}})$.

A ideia é que os modelos com menor *DIC* devem ser preferidos à modelos com maiores valores para essa estatística. Os modelos são penalizados tanto pelo valor de $\hat{D}$, o que favorece um bom ajuste, como também pelo número efetivo de parâmetros p_D .

É importante ressaltar que para o cálculo do *DIC* assume-se que a média a *posteriori* é uma boa estimativa pontual para os parâmetros e, portanto, não representa uma boa medida para a avaliação do ajuste na presença de distribuições a *posteriori* assimétricas para os parâmetros do modelo.

O critério WAIC foi proposto por Watanabe (2010) e segundo Gelman, Hwang e Vehtari (2013), é visto como uma melhoria do critério DIC. O WAIC é calculado por

$$WAIC = -2 \log \int p(\mathbf{y}|\boldsymbol{\theta}) \prod (\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} + \sum_{i=1}^n Var(\log(p(y_i|\boldsymbol{\theta}))). \quad (3.7)$$

O primeiro termo mede o desempenho preditivo, e o segundo termo penaliza a complexidade do modelo. Segundo Gelman, Hwang e Vehtari (2013), o termo complexidade é uma soma da variância para cada ponto de dados separadamente, gerando resultados mais estáveis que o DIC.

A estatística CPO é um método de avaliação cruzada utilizado para avaliar a performance do modelo, medindo o seu poder preditivo (GEISSER; EDDY, 1979). A distribuição preditiva a *posteriori* para a i -ésima observação é

$$p(\mathbf{y}_i|\mathbf{y}_{-i}) = \int p(\mathbf{y}_i|\boldsymbol{\theta}) \prod (\boldsymbol{\theta}|\mathbf{y}_{-i}) d\boldsymbol{\theta},$$

em que $\mathbf{y}_i$ é a i -ésima observação de $\mathbf{y}$ e $\mathbf{y}_{-i}$ é $\mathbf{y}$ sem a observação i . Então, o CPO é a probabilidade a *posteriori* de observar y_i quando o modelo é ajustado sem y_i . Quanto maior o valor da estatística CPO melhor é o ajuste do modelo aos dados. Ao invés de ajustar o modelo a cada vez que é retirada uma observação, uma aproximação do CPO

pode ser obtida por amostragem da distribuição a *posteriori*:

$$CPO_i = \left(\frac{1}{T \sum_{t=1}^T p(\mathbf{y}_i | \boldsymbol{\theta}^t)} \right)^{-1},$$

em que $\boldsymbol{\theta}^t$ é a amostra de $\prod(\boldsymbol{\theta} | \mathbf{y}_i)$ e T indica o número de amostras a *posteriori*.

Ross e Held (2011) indicam o uso da versão “logaritimizada”, sendo definida por $\log CPO = -(n^{-1}) \sum_{i=1}^n \log p(\mathbf{y}_i | \mathbf{y}_{-i})$. Assim, valores mais baixos tanto para DIC quanto logCPO são indicadores de melhor ajuste e qualidade preditiva, respectivamente.

Geralmente, os três critérios não concordam conjuntamente o mesmo modelo, assim, considerou-se primeiro os modelos que apresentaram melhor desempenho baseado no critério logCPO. Nos casos em que os valores do logCPO foram similares entre dois ou mais modelos usou-se o WAIC e, como último critério para escolha do modelo, usou-se o critério DIC.

3.4 Estudos de simulação

Um estudo de simulação de Monte Carlo foi conduzido para avaliar o comportamento sobre amostras finitas dos estimadores dos modelos analisados. Obteve-se resultados de 1.000 simulações de Monte Carlo, realizadas no *software R* (R Core Team, 2018). Em cada réplica, uma amostra aleatória de tamanho n é obtida a partir das distribuições utilizadas neste estudo (Binomial, Poisson, Beta e Gamma) e os parâmetros estimados utilizando inferência Bayesiana por meio do pacote *INLA* (Rue, Martino e Chopin, 2009). Os dados foram gerados do modelo de regressão definido em 3.1, em que $\mathbf{z}_i^t \boldsymbol{\beta} = \beta_0 + \beta_1 z_i$, com z_i gerado de uma distribuição normal padrão. Neste trabalho, realizou-se um estudo de simulação considerando $n = 50, 100$ e 200 , $\beta_0 = 0, 2$ e $\beta_1 = 3$ para a distribuição Beta e $\beta_0 = 7$ e $\beta_1 = 0, 4$ para as demais distribuições consideradas. Para as distribuições Gama e Beta, considerou-se $\phi = 0, 9$.

Para verificar o impacto do método de aproximação para as marginais a *posteriori*, calculou-se, para cada distribuição a probabilidade de cobertura (baseada no intervalo de 95% de confiança), vies e o erro quadrático médio (EQM) para cada um dos parâmetros do modelo.

Para a verificação das probabilidades de cobertura foram construídos intervalos de confiança para os parâmetros. Foram determinados 1.000 intervalos de confiança com um nível de 95%, baseados nas estimativas dos parâmetros para cada configuração de n . Para os 1.000 intervalos, calculou-se quantos continham os verdadeiros valores dos parâmetros para cada configuração.

Os resultados da probabilidade de cobertura são apresentados na Tabela 3.1. Observa-se que o maior impacto na porcentagem de cobertura é na distribuição Binomial. No caso do parâmetro β_0 , as porcentagens tendem a aumentar à medida que o tamanho da amostra aumenta, diferentemente do que é observado no parâmetro β_1 , que tende a diminuir à medida que o tamanho da amostra aumenta. De modo geral, a porcentagem de cobertura para as demais distribuições foram superiores a 92%. Em relação ao método de aproximação utilizado, observa-se que de modo geral, há um pequeno efeito sobre a porcentagem de cobertura, apresentando as maiores diferenças na distribuição Binomial, em que o método simplificado de Laplace apresentou as melhores porcentagens de cobertura. Entretanto, parece não haver um método que supere os outros em termo da porcentagem de cobertura.

Tabela 3.1: Porcentagens de cobertura dos intervalos de credibilidade de 95% para os parâmetros estimados para cada distribuição.

Distribuição	n	método	Parâmetros		
			β_0	β_1	ϕ
Binomial	50	Gaussiano	29,2	99,70	-
		Laplace	27,7	92,30	-
		L. Simplificado	31,2	99,40	-
	100	Gaussiano	47,4	93,70	-
		Laplace	47,2	88,70	-
		L. Simplificado	48,7	93,10	-
	200	Gaussiano	74,6	79,80	-
		Laplace	72,9	74,90	-
		L. Simplificado	76,0	78,30	-
Poisson	50	Gaussiano	96,3	95,70	-
		Laplace	96,2	95,70	-
		L. Simplificado	96,2	95,70	-
	100	Gaussiano	96,1	94,60	-
		Laplace	96,1	94,50	-
		L. Simplificado	96,1	94,60	-
	200	Gaussiano	95,5	95,80	-
		Laplace	95,3	95,80	-
		L. Simplificado	95,3	95,80	-
Beta	50	Gaussiano	94,4	95,00	92,1
		Laplace	94,1	94,90	92,1
		L. Simplificado	94,1	94,90	92,1
	100	Gaussiano	95,0	94,70	93,9
		Laplace	95,0	94,50	93,9
		L. Simplificado	95,0	94,30	93,9
	200	Gaussiano	94,6	95,90	94,9
		Laplace	94,6	95,60	94,9
		L. Simplificado	94,6	95,50	94,9
Gama	50	Gaussiano	94,8	93,70	95,7
		Laplace	95,0	93,80	95,7
		L. Simplificado	95,0	93,70	95,7
	100	Gaussiano	95,3	95,00	94,9
		Laplace	95,9	94,80	94,9
		L. Simplificado	95,6	94,80	94,9
	200	Gaussiano	94,6	95,70	94,9
		Laplace	94,9	95,70	94,9
		L. Simplificado	94,9	95,70	94,9

A Tabela 3.2 apresenta o erro quadrático médio dos parâmetros estimados, para todas as distribuições e diferentes métodos de aproximação para a distribuição condicional do campo latente x . Observa-se novamente que o impacto é maior na distribuição Binomial. De modo geral, o erro quadrático médio dos parâmetros estimados é maior para o método de Laplace, em que apresenta maior discrepância na distribuição Binomial. Note que na distribuição de Poisson o valor do EQM é praticamente nulo. Para as demais distribuições há uma variação mínima entre os métodos.

Tabela 3.2: Erro quadrático médio (EQM) para os parâmetros estimados para cada distribuição.

Distribuição	n	método	Parâmetros		
			β_0	β_1	ϕ
Binomial	50	Gaussiano	17,98576	3,4342093	-
		Laplace	25,97258	5,3868826	-
		L. Simplificado	18,12494	3,5952141	-
	100	Gaussiano	10,78888	1,9457952	-
		Laplace	18,05684	2,3778408	-
		L. Simplificado	10,51810	2,0136035	-
	200	Gaussiano	20,53445	3,9579211	-
		Laplace	42,26462	7,2863166	-
		L. Simplificado	29,37603	5,8056242	-
Poisson	50	Gaussiano	0,000003	0,0000034	-
		Laplace	0,000003	0,0000034	-
		L. Simplificado	0,000003	0,0000034	-
	100	Gaussiano	0,000001	0,0000018	-
		Laplace	0,000001	0,0000018	-
		L. Simplificado	0,000001	0,0000018	-
	200	Gaussiano	0,000001	0,0000007	-
		Laplace	0,000001	0,0000007	-
		L. Simplificado	0,000001	0,0000007	-
Beta	50	Gaussiano	0,029575	0,7647375	0,4477214
		Laplace	0,030136	0,7943613	0,4477212
		L. Simplificado	0,029805	0,7722982	0,4477214
	100	Gaussiano	0,013873	0,3698868	0,1644535
		Laplace	0,014004	0,3750165	0,1644537
		L. Simplificado	0,013932	0,3712362	0,1644535
	200	Gaussiano	0,007072	0,1745105	0,2016067
		Laplace	0,007105	0,1756632	0,2016067
		L. Simplificado	0,007088	0,1748057	0,2016067
Gama	50	Gaussiano	0,005311	0,0058007	0,4902316
		Laplace	0,005298	0,0057953	0,4902316
		L. Simplificado	0,005298	0,0058007	0,4902316
	100	Gaussiano	0,002658	0,0026571	0,2016067
		Laplace	0,002651	0,0026568	0,2016067
		L. Simplificado	0,002653	0,0026576	0,2016067
	200	Gaussiano	0,001357	0,0012883	0,2016067
		Laplace	0,001349	0,0012884	0,2016067
		L. Simplificado	0,001352	0,0012883	0,2016067

Os resultados obtidos nas Tabelas 3.1 e 3.2 mostraram que existe um pequeno impacto nas estimativas dos parâmetros dependendo da escolha do método de aproximação em algumas distribuições.

Os valores médios de DIC, WAIC, logCPO e tempo computacional para cada uma das 1.000 simulações são apresentados na Tabela 3.3. Pode-se observar que todos os critérios de seleção apontam para os modelos ajustados considerando a aproximação de Laplace. Em todas as distribuições, o tempo médio gasto para estimação dos modelos foi

menor quando utilizado o método de aproximação Gaussiano, tal informação corrobora com o artigo publicado por Rue, Martino e Chopin (2009).

Tabela 3.3: Valores médios do DIC, logCPO, WAIC e tempo computacional para diferentes distribuições.

Distribuição	n	método	DIC	logCPO	WAIC	Tempo
Binomial	50	Gaussiano	97,583285	12,761922	20528,548236	3,414
		Laplace	7,193475	9,278332	6,827668	3,425
		L. Simplificado	67,458214	10,786874	7151,930593	3,463
	100	Gaussiano	149,56867	13,879819	34611,665785	4,411
		Laplace	13,95262	8,708700	13,795435	4,739
		L. Simplificado	154,14200	9,433410	53072,270666	4,475
	200	Gaussiano	77,944364	10,467705	63462,599419	4,065
		Laplace	24,042361	4,917956	24,161015	4,251
		L. Simplificado	77,031067	6,296477	28887,149762	4,137
Poisson	50	Gaussiano	569,738400	5,698500	569,716858	3,382
		Laplace	569,737228	5,698503	569,715182	3,396
		L. Simplificado	569,738309	5,698503	569,716870	3,433
	100	Gaussiano	1136,148876	5,680177	1135,931381	3,706
		Laplace	1136,147121	5,680177	1135,929071	3,780
		L. Simplificado	1136,148952	5,680177	1135,931437	3,739
	200	Gaussiano	2272,429492	5,680611	2272,134853	5,275
		Laplace	2272,426257	5,680610	2272,130801	6,661
		L. Simplificado	2272,429521	5,680611	2272,134893	5,380
Beta	50	Gaussiano	20,192775	0,199191	20,030890	4,580
		Laplace	20,134845	0,199092	19,998801	5,309
		L. Simplificado	20,217280	0,199974	20,079535	4,683
	100	Gaussiano	46,393934	0,230825	46,201710	3,733
		Laplace	46,364156	0,230790	46,189256	3,796
		L. Simplificado	46,407998	0,231006	46,230236	3,749
	200	Gaussiano	96,350116	0,240804	96,329268	4,917
		Laplace	96,335382	0,240792	96,323882	5,346
		L. Simplificado	96,356874	0,240842	96,343130	5,013
Gama	50	Gaussiano	762,190738	7,687297	762,625041	8,039
		Laplace	762,035341	7,623897	762,166546	14,899
		L. Simplificado	762,075283	7,632471	762,293691	8,250
	100	Gaussiano	1523,074913	7,619103	1523,253592	6,459
		Laplace	1522,995792	7,615540	1523,035702	9,731
		L. Simplificado	1523,015891	7,616220	1523,091394	6,592
	200	Gaussiano	3038,423118	7,596553	3038,563630	4,875
		Laplace	3038,383992	7,596202	3038,451471	5,313
		L. Simplificado	3038,396021	7,596286	3038,481131	4,860

3.5 Conclusão

Usando um estudo de simulação, os resultados indicaram que o método de aproximação para a distribuição condicional do campo aleatório quase não apresenta impacto nos resultados. No entanto, observa-se um grande impacto da escolha do método quando

é considerada a distribuição Binomial. A abordagem INLA apresentou-se muito eficiente em termos de precisão e custo computacional.

Neste trabalho, focamos em cenários simples para as simulações com o objetivo de investigar o impacto da escolha do método de aproximação para as distribuições condicionais do campo aleatório para os modelos lineares generalizados nas estimativas dos parâmetros. Neste sentido, mais investigações ainda são necessárias para avaliar o comportamento do método em modelos mais complexos, por exemplo, considerando os modelos lineares generalizados mistos.

Nos cenários analisados, tanto em relação as distribuições consideradas quanto ao tamanho das amostras, o método simplificado de Laplace foi o que mais se destacou em termos de precisão e custo computacional.

Referências

DEY, D. K.; GHOSH, S. K.; MALLICK, B. K. *Generalized linear models - a bayesian prospective*. New York: Marcel Dekker, 2000. 423p.

GEISSER, S.; EDDY, W. A predictive approach to model selection. *Journal of the American Statistical Association*, Boston, v. 74, n. 365, p. 153–160, 1979.

GELMAN, A.; HWANG, J; VEHTARI, A. Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, Springer, v. 24, n. 6, p. 997–1016, 2013.

GHOSH, P.; HUANG, L.; YU, B.; TIWARI, R. C. Semiparametric Bayesian approaches to joinpoint regression for population-based cancer survival data. *Computational statistics and data analysis*, Elsevier, v. 53, n. 12, p. 4073–4082, 2009.

MARTINS, T. G.; SIMPSON, D.; LINDGREN, F.; RUE, H. Bayesian computing with INLA: new features. *Computational Statistics and Data Analysis*, Elsevier, v. 67, n. 12, p. 68–93, 2013.

NELDER, J. A.; WEDDERBURN, R. W. M. Generalized Linear Models. *Journal of the Royal Statistical Society Series A*, Wiley for the Royal Statistical Society, v. 135, n. 3, p. 370–384, 1972.

R CORE TEAM (2018). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, <https://www.R-project.org>.

RIZZATO, F. B. *Modelos para análise de dados discretos longitudinais com superdispersão*. 2011. 143p. Tese (Doutorado em Estatística e Experimentação Agronômica) - Escola Superior de Agricultura “Luiz de Queiroz”, Universidade de São Paulo, Piracicaba, 2011.

ROCHA, E. B. *Modelos para a análise de dados de contagem longitudinais com superdispersão: estimação INLA*. 2015. 155p. Tese (Doutorado em Estatística e Experimentação Agronômica) - Escola Superior de Agricultura “Luiz de Queiroz”, Universidade de São Paulo, Piracicaba, 2015.

ROSS, M.; HELD, L. Sensitivity analysis in Bayesian generalized linear mixed models for binary data. *Bayesian Analysis*, v. 6, n. 2, p. 259–278, 2011.

RUE, H.; HELD, L. *Gaussian Markov Random Fields: Theory and Applications*. Chapman & Hall, 2005. 280p.

RUE, H.; MARTINO, S. Approximate Bayesian inference for hierarchical Gaussian Markov random elds models. *Journal of statistical Planning and Inference*, v. 137, p. 3177–3192, 2007.

RUE, H.; MARTINO, S.; CHOPIN, N. Apprximate Bayesian inference for latent Gaussian models by using integrated nested Laplace aproximations (with discussion). *Journal of the Royal Statistical Society–Series B*, Wiley Online Library, v. 71, Part 2, p. 319–392, 2009.

RUE, H.; MARTINO, S.; LINDGREN, F.; SIMPSON, D.; RIEBLER, A. R-inla: Approximate bayesian inference using integrated nested laplace approximations. Tech. rep., Trondheim, Norway (URL <http://www.r-inla.org/>).

SPIEGELHALTER, D. J.; BEST, N. G.; CARLIN, B. P.; LINDE, A. V. D. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society–Series B*, Wiley Online Library, v. 64, n. 4, p. 583–639, 2000.

WATANABE, S. Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, v. 11, n. 2, p. 3571–3594, 2010.

APÊNDICES

Apêndice A: R capítulo 2

Realizou-se toda a programação no R. Foram utilizados os seguintes códigos:

```
# Pacotes
rm(list=ls(all=TRUE))
library(INLA)
library(xtable)
library(gridExtra)
library(ggplot2)
library(geoR)
library(dplyr)
library(fields)
library(lattice)
library(splancs)

# Inicio da analise

n_y      <- 200
limits   <- matrix(cbind(c(0,1,1,1,0), c(0,0,1,1,1)),ncol=2)
loc_y    <- cbind(runif(n_y), runif(n_y))
beta0    <- 10
beta1    <- 5
kappa_xi <- 5
kappa_u  <- 7
sigma2_xi <- 0.5
sigma2_u  <- 0.3
pts      <- loc_y
dmat     <- dist(pts)
mesh1    <- inla.mesh.2d(loc = pts, max.edge = 0.15, cutoff = 0.03, offset = 0.1)

j        <- 100
Beta0.1  <- matrix(NA,ncol=5,nrow=j)
Beta0.2  <- matrix(NA,ncol=5,nrow=j)
Beta0.3  <- matrix(NA,ncol=5,nrow=j)

Sig2U1   <- matrix(NA,ncol=3,nrow=j)
Sig2U2   <- matrix(NA,ncol=3,nrow=j)
```

```

Sig2U3      <- matrix(NA,ncol=3,nrow=j)

RangeU1     <- matrix(NA,ncol=3,nrow=j)
RangeU2     <- matrix(NA,ncol=3,nrow=j)
RangeU3     <- matrix(NA,ncol=3,nrow=j)

RMSE1       <- matrix(NA,ncol=1,nrow=j)
RMSE2       <- matrix(NA,ncol=1,nrow=j)
RMSE3       <- matrix(NA,ncol=1,nrow=j)

DIC1        <- matrix(NA,ncol=1,nrow=j)
DIC2        <- matrix(NA,ncol=1,nrow=j)
DIC3        <- matrix(NA,ncol=1,nrow=j)

CP01        <- matrix(NA,ncol=1,nrow=j)
CP02        <- matrix(NA,ncol=1,nrow=j)
CP03        <- matrix(NA,ncol=1,nrow=j)

WAIC1       <- matrix(NA,ncol=1,nrow=j)
WAIC2       <- matrix(NA,ncol=1,nrow=j)
WAIC3       <- matrix(NA,ncol=1,nrow=j)

Times1      <- matrix(NA,ncol=1,nrow=j)
Times2      <- matrix(NA,ncol=1,nrow=j)
Times3      <- matrix(NA,ncol=1,nrow=j)

simulate_GF <- function(coords, kappa, variance, lambda=1) {
  n      <- nrow(coords)
  dist.m <- as.matrix(dist(coords))
  cor.m  <- 2^(1-lambda)/gamma(lambda)*(kappa*dist.m)^lambda*
  besselK(x=dist.m*kappa,nu=lambda)
  diag(cor.m) <- 1
  Sigma <- variance * cor.m
  c(chol(Sigma) %*% rnorm(n=n,mean=0,sd=1))
}

pb      <- txtProgressBar(min = 0, max = j, style = 3)
for(i in 35:j){
  Sys.sleep(0.1)

```

```

setTxtProgressBar(pb, i)
cat("\n\n rep = ",i,"\n\n")
u   <- simulate_GF(coords=loc_y, kappa=kappa_u, variance=sigma2_u)
ntrials   <- 5 + rpois(n=n_y, lambda=10)
eta_y     <- beta0 + ## beta1 * xi[n_x + 1:n_y]
y1        <- rbinom(n=n_y, size=ntrials, prob=exp(eta_y)/(1 + exp(eta_y)) )
data      <- data.frame(pts,y1)
strate    <- c("simplified.laplace","gaussian","laplace")
lres <- index.val <- u_field <- list()
for(k in 1:3){
cat("\n\n modelo",k,"\n\n")
l.spde    <- inla.spde2.matern(mesh = mesh1, alpha=2)
l.A_y     <- inla.spde.make.A(mesh = mesh1, loc=loc_y)
s.index    <- inla.spde.make.index(name = "spatial.field", n.spde = l.spde$n.spde)
stk_y     <- inla.stack(data      = list(y = y1, Ntrials=ntrials,
link="logit"),
effects = list(u.field=1:l.spde$n.spde,
intercept=rep(1,length(y1))),
A       = list(l.A_y, 1),
tag     = "est.y")
stk      <- inla.stack(stk_y)
formula  <- y ~ -1 + intercept +
f(u.field, model=l.spde)
lres[[k]] <- inla(formula,
data      = inla.stack.data(stk),
family    = "binomial",
Ntrials   = inla.stack.data(stk)$Ntrials,
control.predictor = list(A=inla.stack.A(stk),
compute=TRUE),
control.inla = list(strategy = strate[k]),
control.compute = list(dic=TRUE,cpo=TRUE,waic=TRUE))
u_field[[k]] <- inla.spde2.result(lres[[k]], name="u.field", l.spde)
index.val[[k]] <- inla.stack.index(stk,"est.y")$data
}

Beta0    <- t(apply(lres, function(m) m$summary.fixed[,1:5]))
Beta0

Sig2U    <- lapply(u_field, function(m) m$marginals.variance.nominal[[1]])

```

```

Sig2U_m <- sapply(Sig2U, function(m) inla.emarginal(function(x) x, m))
Sig2U_qt <- sapply(Sig2U, function(m) inla.hpdmarginal(.95, m))
Sig2U_t <- data.frame(Sig2U_m, Sig2U_qt[1], Sig2U_qt[2])
dimnames(Sig2U_t)[2] <- list(c("mean", "low", "high"))

RangeU <- lapply(u_field, function(m) m$marginals.range[[1]])
RangeU_m <- sapply(RangeU, function(m) inla.emarginal(function(x) x, m))
RangeU_qt <- sapply(RangeU, function(m) inla.hpdmarginal(.95, m))
RangeU_t <- data.frame(RangeU_m, RangeU_qt[1], RangeU_qt[2])
dimnames(RangeU_t)[2] <- list(c("mean", "low", "high"))

Beta0.1[i,] <- unlist(Beta0[1,])
Beta0.2[i,] <- unlist(Beta0[2,])
Beta0.3[i,] <- unlist(Beta0[3,])

Sig2U1[i,] <- unlist(Sig2U_t[1,])
Sig2U2[i,] <- unlist(Sig2U_t[2,])
Sig2U3[i,] <- unlist(Sig2U_t[3,])

RangeU1[i,] <- unlist(RangeU_t[1,])
RangeU2[i,] <- unlist(RangeU_t[2,])
RangeU3[i,] <- unlist(RangeU_t[3,])

DIC <- sapply(lres, function(m) m$dic$dic)
DIC1[i,] <- DIC[1]
DIC2[i,] <- DIC[2]
DIC3[i,] <- DIC[3]

CP0 <- sapply(lres, function(m) mean(log(m$cpo$cpo), na.rm=TRUE))
CP01[i,] <- CP0[1]
CP02[i,] <- CP0[2]
CP03[i,] <- CP0[3]

WAIC <- sapply(lres, function(m) m$waic$waic)
WAIC1[i,] <- WAIC[1]
WAIC2[i,] <- WAIC[2]
WAIC3[i,] <- WAIC[3]

Times <- sapply(lres, function(m) m$cpu[4]/60)

```

```

Times1[i,] <- Times[1]
Times2[i,] <- Times[2]
Times3[i,] <- Times[3]
}

```

```

BBeta0 <- paste(rep("Beta0.",3),1:3,sep="")
SSig2U <- paste(rep("Sig2U",3),1:3,sep="")
RRangeU <- paste(rep("RangeU",3),1:3,sep="")
DDIC <- paste(rep("DIC",3),1:3,sep="")
CCP0 <- paste(rep("CP0",3),1:3,sep="")
WWAIC <- paste(rep("WAIC",3),1:3,sep="")
TTimes <- paste(rep("Times",3),1:3,sep="")

```

```

for(w in 1:3){
strate <- c("simplified_laplace","gaussian","laplace")
write.table(get(BBeta0[w]) ,
paste("B0_",strate[w],"_100.txt",sep=""), row.names=FALSE)
write.table(get(SSig2U[w]) ,
paste("Sig2U_",strate[w],"_100.txt",sep=""), row.names=FALSE)
write.table(get(RRangeU[w]) ,
paste("RangeU_",strate[w],"_100.txt",sep=""), row.names=FALSE)
write.table(get(DDIC[w]) ,
paste("DIC_",strate[w],"_100.txt",sep=""), row.names=FALSE)
write.table(get(CCP0[w]) ,
paste("CP0_",strate[w],"_100.txt",sep=""), row.names=FALSE)
write.table(get(WWAIC[w]) ,
paste("WAIC_",strate[w],"_100.txt",sep=""), row.names=FALSE)
write.table(get(TTimes[w]) ,
paste("Times_",strate[w],"_100.txt",sep=""), row.names=FALSE)
}

```

```

#####
#####
strate <- c("simplified_laplace","gaussian","laplace")
beta0s <- c(paste("B0_",
c("simplified_laplace","gaussian","laplace"),
rep(c("_100")),sep=''))
Sig2Us <- c(paste("Sig2U_",
c("simplified_laplace","gaussian","laplace"),

```



```

rep(c("_100")),sep='')
RangeUs  <- c(paste("RangeU_",
c("simplified_laplace","gaussian","laplace"),
rep(c("_100")),sep=''))
DICs     <- c(paste("DIC_",
c("simplified_laplace","gaussian","laplace"),
rep(c("_100")),sep=''))
CPOs     <- c(paste("CPO_",
c("simplified_laplace","gaussian","laplace"),
rep(c("_100")),sep=''))
WAICs    <- c(paste("WAIC_",
c("simplified_laplace","gaussian","laplace"),
rep(c("_100")),sep=''))
Timess   <- c(paste("Times_",
c("simplified_laplace","gaussian","laplace"),
rep(c("_100")),sep=''))

namesPara <- c("beta0s","Sig2Us","RangeUs")

coverage <- matrix(NA,ncol=length(namesPara),nrow=length(beta0s))
dimnames(coverage)[1] <- list(paste(c("simplified_laplace","gaussian","laplace"),
rep(c("_100")),sep=''))
dimnames(coverage)[2] <- list(namesPara)

Vies     <- matrix(NA,ncol=length(namesPara),nrow=length(beta0s))
dimnames(Vies)[2] <- list(namesPara)
dimnames(Vies)[1] <- list(paste(c("simplified_laplace","gaussian","laplace"),
rep(c("_100")),sep=''))
EQM      <- matrix(NA,ncol=length(namesPara),nrow=length(beta0s))
dimnames(EQM)[2] <- list(namesPara)
dimnames(EQM)[1] <- list(paste(c("simplified_laplace","gaussian","laplace"),
rep(c("_100")),sep=''))

true     <- c(beta0,sigma2_u,sqrt(8)/kappa_u)
for(k in 1:length(namesPara)){
  for(j in 1:length(beta0s)){
    Analise <- read.table(paste(get(noquote(namesPara)[k]),
".txt",sep=""),[j],h=TRUE)
    if(ncol(Analise)==5) Analise <- Analise[,-c(2,4)]
  }
}

```

```

colnames(Analise) <- c("mean","low","high")
head(Analise)
xA=matrix(NA,ncol=1,nrow=nrow(Analise))
for(i in 1:nrow(Analise)){
  if((Analise[i,2]>true[k] || Analise[i,3]<true[k]) == "TRUE")
  xA[i,]<-1 else xA[i,]<-0
}
cont = apply(xA,2,sum)/(nrow(xA))
coverage[j,k] <- (1-cont)*100
Vies[j,k]      <- mean(Analise[,1])-true[k]
EQM[j,k]       <- var(Analise[,1]) + Vies[j,k]^2
}
}
coverage
abs(Vies)
EQM

outrosNomes <- c("RMSEs","DICs","CPOs","WAICs","Timess")

OUTROS      <- matrix(NA,ncol=length(outrosNomes),nrow=length(beta0s))
dimnames(OUTROS)[2] <- list(outrosNomes)
dimnames(OUTROS)[1] <- list(paste(c(rep("mesh",14)),
1:14,
rep(c("_50","_100","_200"),rep(14,3)),sep=' '))
for(k in 1:length(outrosNomes)){
  for(j in 1:length(DICs)){
    OUTRO    <- read.table(paste(get(noquote(outrosNomes)[k]),
".txt",sep="")[j],h=TRUE)
    OUTROS[j,k] <- mean(abs(OUTRO$V1))
  }}
OUTROS

```

Apêndice A: R capítulo 3

Realizou-se toda a programação no R. Foram utilizados os seguintes códigos:

```

rm(list=ls(all=TRUE))
library(INLA)
library(xtable)
library(gridExtra)

```

```
library(ggplot2)
library(geoR)
library(dplyr)
library(fields)
library(lattice)
library(splancs)

# Parametros
par_poisson    = c(7,.4)
par_binomial   = c(7,.4)
par_T          = c(7,.4,6)
par_beta       = c(.2,3,3)
par_gamma      = c(7,.4,3)

# Numero de simulacoes
j              <- 1000

Beta0.1_P <- matrix(NA,ncol=5,nrow=j)
Beta0.2_P <- matrix(NA,ncol=5,nrow=j)
Beta0.3_P <- matrix(NA,ncol=5,nrow=j)

Beta1.1_P <- matrix(NA,ncol=5,nrow=j)
Beta1.2_P <- matrix(NA,ncol=5,nrow=j)
Beta1.3_P <- matrix(NA,ncol=5,nrow=j)

Beta0.1_B <- matrix(NA,ncol=5,nrow=j)
Beta0.2_B <- matrix(NA,ncol=5,nrow=j)
Beta0.3_B <- matrix(NA,ncol=5,nrow=j)

Beta1.1_B <- matrix(NA,ncol=5,nrow=j)
Beta1.2_B <- matrix(NA,ncol=5,nrow=j)
Beta1.3_B <- matrix(NA,ncol=5,nrow=j)

Beta0.1_Be <- matrix(NA,ncol=5,nrow=j)
Beta0.2_Be <- matrix(NA,ncol=5,nrow=j)
Beta0.3_Be <- matrix(NA,ncol=5,nrow=j)

Beta1.1_Be <- matrix(NA,ncol=5,nrow=j)
Beta1.2_Be <- matrix(NA,ncol=5,nrow=j)
```

```
Beta1.3_Be <- matrix(NA,ncol=5,nrow=j)
```

```
Beta0.1_G <- matrix(NA,ncol=5,nrow=j)
```

```
Beta0.2_G <- matrix(NA,ncol=5,nrow=j)
```

```
Beta0.3_G <- matrix(NA,ncol=5,nrow=j)
```

```
Beta1.1_G <- matrix(NA,ncol=5,nrow=j)
```

```
Beta1.2_G <- matrix(NA,ncol=5,nrow=j)
```

```
Beta1.3_G <- matrix(NA,ncol=5,nrow=j)
```

```
prec.1_beta <- matrix(NA,ncol=5,nrow=j)
```

```
prec.2_beta <- matrix(NA,ncol=5,nrow=j)
```

```
prec.3_beta <- matrix(NA,ncol=5,nrow=j)
```

```
prec.1_gamma <- matrix(NA,ncol=5,nrow=j)
```

```
prec.2_gamma <- matrix(NA,ncol=5,nrow=j)
```

```
prec.3_gamma <- matrix(NA,ncol=5,nrow=j)
```

```
DIC_P_1 <- matrix(NA,ncol=1,nrow=j)
```

```
DIC_P_2 <- matrix(NA,ncol=1,nrow=j)
```

```
DIC_P_3 <- matrix(NA,ncol=1,nrow=j)
```

```
DIC_B_1 <- matrix(NA,ncol=1,nrow=j)
```

```
DIC_B_2 <- matrix(NA,ncol=1,nrow=j)
```

```
DIC_B_3 <- matrix(NA,ncol=1,nrow=j)
```

```
DIC_Be_1 <- matrix(NA,ncol=1,nrow=j)
```

```
DIC_Be_2 <- matrix(NA,ncol=1,nrow=j)
```

```
DIC_Be_3 <- matrix(NA,ncol=1,nrow=j)
```

```
DIC_G_1 <- matrix(NA,ncol=1,nrow=j)
```

```
DIC_G_2 <- matrix(NA,ncol=1,nrow=j)
```

```
DIC_G_3 <- matrix(NA,ncol=1,nrow=j)
```

```
CPO_P_1 <- matrix(NA,ncol=1,nrow=j)
```

```
CPO_P_2 <- matrix(NA,ncol=1,nrow=j)
```

```
CPO_P_3 <- matrix(NA,ncol=1,nrow=j)
```

```
CPO_B_1 <- matrix(NA,ncol=1,nrow=j)
```

```
CPO_B_2 <- matrix(NA,ncol=1,nrow=j)
CPO_B_3 <- matrix(NA,ncol=1,nrow=j)

CPO_Be_1 <- matrix(NA,ncol=1,nrow=j)
CPO_Be_2 <- matrix(NA,ncol=1,nrow=j)
CPO_Be_3 <- matrix(NA,ncol=1,nrow=j)

CPO_G_1 <- matrix(NA,ncol=1,nrow=j)
CPO_G_2 <- matrix(NA,ncol=1,nrow=j)
CPO_G_3 <- matrix(NA,ncol=1,nrow=j)

WAIC_P_1 <- matrix(NA,ncol=1,nrow=j)
WAIC_P_2 <- matrix(NA,ncol=1,nrow=j)
WAIC_P_3 <- matrix(NA,ncol=1,nrow=j)

WAIC_B_1 <- matrix(NA,ncol=1,nrow=j)
WAIC_B_2 <- matrix(NA,ncol=1,nrow=j)
WAIC_B_3 <- matrix(NA,ncol=1,nrow=j)

WAIC_Be_1 <- matrix(NA,ncol=1,nrow=j)
WAIC_Be_2 <- matrix(NA,ncol=1,nrow=j)
WAIC_Be_3 <- matrix(NA,ncol=1,nrow=j)

WAIC_G_1 <- matrix(NA,ncol=1,nrow=j)
WAIC_G_2 <- matrix(NA,ncol=1,nrow=j)
WAIC_G_3 <- matrix(NA,ncol=1,nrow=j)

Times_P_1 <- matrix(NA,ncol=1,nrow=j)
Times_P_2 <- matrix(NA,ncol=1,nrow=j)
Times_P_3 <- matrix(NA,ncol=1,nrow=j)

Times_B_1 <- matrix(NA,ncol=1,nrow=j)
Times_B_2 <- matrix(NA,ncol=1,nrow=j)
Times_B_3 <- matrix(NA,ncol=1,nrow=j)

Times_Be_1 <- matrix(NA,ncol=1,nrow=j)
Times_Be_2 <- matrix(NA,ncol=1,nrow=j)
Times_Be_3 <- matrix(NA,ncol=1,nrow=j)
```

```

Times_G_1 <- matrix(NA,ncol=1,nrow=j)
Times_G_2 <- matrix(NA,ncol=1,nrow=j)
Times_G_3 <- matrix(NA,ncol=1,nrow=j)

pb          <- txtProgressBar(min = 0, max = j, style = 3)
for(i in 977:j){
  Sys.sleep(0.1)
  setTxtProgressBar(pb, i)
  cat("\n\n rep = ",i,"\n\n")
#####
#####
#####
  # Valores aleatorios
  n          = 200
  x          = rnorm(n)
  z          = rnorm(n, sd=0.2)
  w          = runif(n, min = 0.25, max = 0.75)

  # Etas
  eta_poisson = par_poisson[1] + par_poisson[2]*x
  eta_binomial = par_binomial[1] + par_binomial[2]*x
  eta_beta    = par_beta[1] + par_beta[2]*z
  eta_gamma   = par_gamma[1] + par_gamma[2]*x

  # Outros valores
  E          = sample(1:10, n, replace=TRUE)
  lambda     = E*exp(eta_poisson)
  Ntrials    = sample(c(1,5,10,15), size=n, replace=TRUE)
  prob       = exp(eta_binomial)/(1 + exp(eta_binomial))
  phi        = par_beta[3] * w
  mu_beta    = exp(eta_beta)/(1+exp(eta_beta))
  a_beta     = mu_beta * phi
  b_beta     = -mu_beta * phi + phi
  mu_gamma   = exp(eta_gamma)
  prec.scale = runif(n, min = 0.5, max = 2)
  prec.par   = par_gamma[3]
  a_gamma    = prec.par * prec.scale
  b_gamma    = mu_gamma / (prec.par * prec.scale)

```

```

# Amostras
y_poisson      = rpois(n, lambda = lambda)
y_binomial     = rbinom(n, size=Ntrials, prob = prob)
y_beta         = rbeta(n, a_beta, b_beta)
y_gamma        = rgamma(n, shape = a_gamma, scale = b_gamma)

# Datas
data_poisson   = list(y=y_poisson,x=x)
data_binomial  = list(y=y_binomial,x=x)
data_beta      = data.frame(y=y_beta, z=z, w=w)
data_gamma     = data.frame(y=y_gamma,x=x)

# Formulas
formula        = y ~ 1 + x
formula_beta   = y ~ 1 + z

# Metodos
metodos <- c("simplified.laplace","gaussian","laplace")

# Testado os metodos de aproximacao
lres_poisson <- lres_binomial <- lres_beta <- lres_gamma <- list()
for(k in 1:3){
  cat("\n\n modelo",k,"\n\n")
  # INLA
  lres_poisson[[k]] =
  inla(formula, family = "poisson", data = data_poisson, E=E,
  control.inla = list(strategy=metodos[k]),
  control.compute = list(dic=TRUE,cpo=TRUE,waic=TRUE))
  lres_binomial[[k]] =
  inla(formula, family = "binomial", data = data_binomial, Ntrials=Ntrials,
  control.inla = list(strategy=metodos[k]),
  control.compute = list(dic=TRUE,cpo=TRUE,waic=TRUE))
  lres_beta[[k]] =
  inla(formula_beta, family = "beta", data = data_beta, scale = w,
  control.inla = list(strategy=metodos[k]),
  control.compute = list(dic=TRUE,cpo=TRUE,waic=TRUE))
  lres_gamma[[k]] =
  inla(formula, family = "gamma", data = data_gamma, scale = prec.scale,

```

```

control.inla = list(strategy=metodos[k]),
control.compute = list(dic=TRUE,cpo=TRUE,waic=TRUE))
}
# Poisson
Beta0_P    <- t(sapply(lres_poisson, function(m) m$summary.fixed[1,1:5]))
Beta0_P

Beta0.1_P[i,] <- unlist(Beta0_P[1,])
Beta0.2_P[i,] <- unlist(Beta0_P[2,])
Beta0.3_P[i,] <- unlist(Beta0_P[3,])

Beta1_P    <- t(sapply(lres_poisson, function(m) m$summary.fixed[2,1:5]))
Beta1_P

Beta1.1_P[i,] <- unlist(Beta1_P[1,])
Beta1.2_P[i,] <- unlist(Beta1_P[2,])
Beta1.3_P[i,] <- unlist(Beta1_P[3,])

# Binomial
Beta0_B    <- t(sapply(lres_binomial, function(m) m$summary.fixed[1,1:5]))
Beta0_B

Beta0.1_B[i,] <- unlist(Beta0_B[1,])
Beta0.2_B[i,] <- unlist(Beta0_B[2,])
Beta0.3_B[i,] <- unlist(Beta0_B[3,])

Beta1_B    <- t(sapply(lres_binomial, function(m) m$summary.fixed[2,1:5]))
Beta1_B

Beta1.1_B[i,] <- unlist(Beta1_B[1,])
Beta1.2_B[i,] <- unlist(Beta1_B[2,])
Beta1.3_B[i,] <- unlist(Beta1_B[3,])

# Beta
Beta0_Be    <- t(sapply(lres_beta, function(m) m$summary.fixed[1,1:5]))
Beta0_Be

Beta0.1_Be[i,] <- unlist(Beta0_Be[1,])
Beta0.2_Be[i,] <- unlist(Beta0_Be[2,])

```



```

Beta0.3_Be[i,] <- unlist(Beta0_Be[3,])

Beta1_Be <- t(sapply(lres_beta, function(m) m$summary.fixed[2,1:5]))
Beta1_Be

Beta1.1_Be[i,] <- unlist(Beta1_Be[1,])
Beta1.2_Be[i,] <- unlist(Beta1_Be[2,])
Beta1.3_Be[i,] <- unlist(Beta1_Be[3,])

preci.beta <- t(sapply(lres_beta, function(m) m$summary.hyperpar[1,1:5]))
preci.beta

prec.1_beta[i,] <- unlist(preci.beta[1,])
prec.2_beta[i,] <- unlist(preci.beta[2,])
prec.3_beta[i,] <- unlist(preci.beta[3,])

# Gamma
Beta0_G <- t(sapply(lres_gamma, function(m) m$summary.fixed[1,1:5]))
Beta0_G

Beta0.1_G[i,] <- unlist(Beta0_G[1,])
Beta0.2_G[i,] <- unlist(Beta0_G[2,])
Beta0.3_G[i,] <- unlist(Beta0_G[3,])

Beta1_G <- t(sapply(lres_gamma, function(m) m$summary.fixed[2,1:5]))
Beta1_G

Beta1.1_G[i,] <- unlist(Beta1_G[1,])
Beta1.2_G[i,] <- unlist(Beta1_G[2,])
Beta1.3_G[i,] <- unlist(Beta1_G[3,])

preci.gamma <- t(sapply(lres_gamma, function(m) m$summary.hyperpar[1,1:5]))
preci.gamma

prec.1_gamma[i,] <- unlist(preci.gamma[1,])
prec.2_gamma[i,] <- unlist(preci.gamma[2,])
prec.3_gamma[i,] <- unlist(preci.gamma[3,])

DIC_P <- sapply(lres_poisson, function(m) m$dic$dic)

```

```

DIC_B      <- sapply(lres_binomial, function(m) m$dic$dic)
DIC_Be     <- sapply(lres_beta, function(m) m$dic$dic)
DIC_G      <- sapply(lres_gamma, function(m) m$dic$dic)

DIC_P_1[i,] <- DIC_P[1]
DIC_P_2[i,] <- DIC_P[2]
DIC_P_3[i,] <- DIC_P[3]

DIC_B_1[i,] <- DIC_B[1]
DIC_B_2[i,] <- DIC_B[2]
DIC_B_3[i,] <- DIC_B[3]

DIC_Be_1[i,] <- DIC_Be[1]
DIC_Be_2[i,] <- DIC_Be[2]
DIC_Be_3[i,] <- DIC_Be[3]

DIC_G_1[i,] <- DIC_G[1]
DIC_G_2[i,] <- DIC_G[2]
DIC_G_3[i,] <- DIC_G[3]

CPO_P      <- sapply(lres_poisson, function(m) -mean(log(m$cpo$cpo),na.rm=TRUE))
CPO_B      <- sapply(lres_binomial, function(m) -mean(log(m$cpo$cpo),na.rm=TRUE))
CPO_Be     <- sapply(lres_beta, function(m) -mean(log(m$cpo$cpo),na.rm=TRUE))
CPO_G      <- sapply(lres_gamma, function(m) -mean(log(m$cpo$cpo),na.rm=TRUE))

CPO_P_1[i,] <- CPO_P[1]
CPO_P_2[i,] <- CPO_P[2]
CPO_P_3[i,] <- CPO_P[3]

CPO_B_1[i,] <- CPO_B[1]
CPO_B_2[i,] <- CPO_B[2]
CPO_B_3[i,] <- CPO_B[3]

CPO_Be_1[i,] <- CPO_Be[1]
CPO_Be_2[i,] <- CPO_Be[2]
CPO_Be_3[i,] <- CPO_Be[3]

CPO_G_1[i,] <- CPO_G[1]
CPO_G_2[i,] <- CPO_G[2]

```

```

CPO_G_3[i,]    <- CPO_G[3]

WAIC_P        <- sapply(lres_poisson, function(m) m$waic$waic)
WAIC_B        <- sapply(lres_binomial, function(m) m$waic$waic)
WAIC_Be       <- sapply(lres_beta, function(m) m$waic$waic)
WAIC_G        <- sapply(lres_gamma, function(m) m$waic$waic)

WAIC_P_1[i,]   <- WAIC_P[1]
WAIC_P_2[i,]   <- WAIC_P[2]
WAIC_P_3[i,]   <- WAIC_P[3]

WAIC_B_1[i,]   <- WAIC_B[1]
WAIC_B_2[i,]   <- WAIC_B[2]
WAIC_B_3[i,]   <- WAIC_B[3]

WAIC_Be_1[i,]  <- WAIC_Be[1]
WAIC_Be_2[i,]  <- WAIC_Be[2]
WAIC_Be_3[i,]  <- WAIC_Be[3]

WAIC_G_1[i,]   <- WAIC_G[1]
WAIC_G_2[i,]   <- WAIC_G[2]
WAIC_G_3[i,]   <- WAIC_G[3]

Times_P        <- sapply(lres_poisson, function(m) m$cpu[4]/60)
Times_B        <- sapply(lres_binomial, function(m) m$cpu[4]/60)
Times_Be       <- sapply(lres_beta, function(m) m$cpu[4]/60)
Times_G        <- sapply(lres_gamma, function(m) m$cpu[4]/60)

Times_P_1[i,]  <- Times_P[1]
Times_P_2[i,]  <- Times_P[2]
Times_P_3[i,]  <- Times_P[3]

Times_B_1[i,]  <- Times_B[1]
Times_B_2[i,]  <- Times_B[2]
Times_B_3[i,]  <- Times_B[3]

Times_Be_1[i,] <- Times_Be[1]
Times_Be_2[i,] <- Times_Be[2]
Times_Be_3[i,] <- Times_Be[3]

```

```

Times_G_1[i,]    <- Times_G[1]
Times_G_2[i,]    <- Times_G[2]
Times_G_3[i,]    <- Times_G[3]
}

BBeta0  <- paste(rep("Beta0.",12),1:3,rep(c("_P","_B","_Be","_G"),rep(3,4)),sep="")
BBeta1  <- paste(rep("Beta1.",12),1:3,rep(c("_P","_B","_Be","_G"),rep(3,4)),sep="")
Pprec   <- paste(rep("prec.",6),1:3,rep(c("_beta","_gamma"),rep(3,2)),sep="")
DDIC    <- paste(rep("DIC_",12),rep(c("P_", "B_", "Be_", "G_"),rep(3,4)),1:3,sep="")
CCPO    <- paste(rep("CPO_",12),rep(c("P_", "B_", "Be_", "G_"),rep(3,4)),1:3,sep="")
WWAIC   <- paste(rep("WAIC_",12),rep(c("P_", "B_", "Be_", "G_"),rep(3,4)),1:3,sep="")
TTimes  <- paste(rep("Times_",12),rep(c("P_", "B_", "Be_", "G_"),rep(3,4)),1:3,sep="")

NOMES <- paste(rep(c("P_", "B_", "Be_", "G_"),rep(3,4)),
c("simplified_laplace", "gaussian", "laplace"),
sep="")

for(j in 1:12){
write.table(get(BBeta0[j]) ,paste("B0_",NOMES[j], "_200.txt",sep=""), row.names=FALSE)
write.table(get(BBeta1[j]) ,paste("B1_",NOMES[j], "_200.txt",sep=""), row.names=FALSE)
write.table(get(DDIC[j]) ,paste("DIC_",NOMES[j], "_200.txt",sep=""), row.names=FALSE)
write.table(get(CCPO[j]) ,paste("CPO_",NOMES[j], "_200.txt",sep=""), row.names=FALSE)
write.table(get(WWAIC[j]) ,paste("WAIC_",NOMES[j], "_200.txt",sep=""), row.names=FALSE)
write.table(get(TTimes[j]) ,
paste("Times_",NOMES[j], "_200.txt",sep=""), row.names=FALSE)
}

NOMES2 <- paste(rep(c("Be_", "G_"),rep(3,2)),
c("simplified_laplace", "gaussian", "laplace"),
sep="")
for(w in 1:6){
write.table(get(Pprec[w]) ,
paste("prec_",NOMES2[w], "_200.txt",sep=""), row.names=FALSE)
}

```

```

#####
#####

```

```

# Parametros
par_poisson    = c(7,.4)
par_binomial   = c(7,.4)
par_beta       = c(.2,3,3)
par_gamma      = c(7,.4,3)

beta0s    <- c(paste("B0_",
rep(c("B_", "Be_", "G_", "P_"), rep(3,4)),
c("gaussian", "laplace", "simplified_laplace"),
rep(c("_50", "_100", "_200"), rep(12,3)), sep=''))
beta1s    <- c(paste("B1_",
rep(c("B_", "Be_", "G_", "P_"), rep(3,4)),
c("gaussian", "laplace", "simplified_laplace"),
rep(c("_50", "_100", "_200"), rep(12,3)), sep=''))
precS     <- c(paste("prec_",
rep(c("Be_", "G_"), rep(3,2)),
c("gaussian", "laplace", "simplified_laplace"),
rep(c("_50", "_100", "_200"), rep(6,3)), sep=''))
DICs       <- c(paste("DIC_",
rep(c("B_", "Be_", "G_", "P_"), rep(3,4)),
c("gaussian", "laplace", "simplified_laplace"),
rep(c("_50", "_100", "_200"), rep(12,3)), sep=''))
CPOs       <- c(paste("CPO_",
rep(c("B_", "Be_", "G_", "P_"), rep(3,4)),
c("gaussian", "laplace", "simplified_laplace"),
rep(c("_50", "_100", "_200"), rep(12,3)), sep=''))
WAICs      <- c(paste("WAIC_",
rep(c("B_", "Be_", "G_", "P_"), rep(3,4)),
c("gaussian", "laplace", "simplified_laplace"),
rep(c("_50", "_100", "_200"), rep(12,3)), sep=''))
TimesS     <- c(paste("Times_",
rep(c("B_", "Be_", "G_", "P_"), rep(3,4)),
c("gaussian", "laplace", "simplified_laplace"),
rep(c("_50", "_100", "_200"), rep(12,3)), sep=''))

namesParabetas <- c("beta0s", "beta1s")

coverage <- matrix(NA, ncol=3, nrow=length(beta0s))

```

```

dimnames(coverage)[1] <-
list(paste(rep(c("Binomial_", "Beta_", "Gama_", "Poisson_"), rep(3,4)),
c("gaussian", "laplace", "simplified_laplace"),
rep(c("_50", "_100", "_200"), rep(12,3)), sep=''))
dimnames(coverage)[2] <- list(c("Beta0", "Beta1", "Precisao"))

Vies <- matrix(NA, ncol=3, nrow=length(beta0s))
dimnames(Vies)[2] <- list(c("Beta0", "Beta1", "Precisao"))
dimnames(Vies)[1] <- list(paste(rep(c("Binomial_", "Beta_", "Gama_", "Poisson_"), rep(3,4)),
c("gaussian", "laplace", "simplified_laplace"),
rep(c("_50", "_100", "_200"), rep(12,3)), sep=''))
EQM <- matrix(NA, ncol=3, nrow=length(beta0s))
dimnames(EQM)[2] <- list(c("Beta0", "Beta1", "Precisao"))
dimnames(EQM)[1] <- list(paste(rep(c("Binomial_", "Beta_", "Gama_", "Poisson_"), rep(3,4)),
c("gaussian", "laplace", "simplified_laplace"),
rep(c("_50", "_100", "_200"), rep(12,3)), sep=''))

for(k in 1:length(namesParabetas)){
  for(j in 1:length(beta0s)){
    Analise <- na.omit(read.table(paste(get(noquote(namesParabetas)[k]),
".txt", sep=""))[j], h=TRUE))
    if(ncol(Analise)==5) Analise <- Analise[, -c(2,4)]
    colnames(Analise) <- c("mean", "low", "high")
    head(Analise)
    if(j==1||j==2||j==3|| j==13||j==14||j==15|| j==25||j==26||j==27){
      true <- par_binomial
    }else if(j==4||j==5||j==6|| j==16||j==17||j==18|| j==28||j==29||j==30){
      true <- par_beta
    }else if(j==7||j==8||j==9|| j==19||j==20||j==21|| j==31||j==32||j==33){
      true <- par_gamma
    }else if(j==10||j==11||j==12|| j==22||j==23||j==24|| j==34||j==35||j==36){
      true <- par_poisson
    }
    xA=matrix(NA, ncol=1, nrow=nrow(Analise))
    for(i in 1:nrow(Analise)){
      if((Analise[i,2]>true[k] || Analise[i,3]<true[k]) == "TRUE")
      xA[i,]<-1 else xA[i,]<-0
    }
    cont = apply(xA, 2, sum)/(nrow(xA))
  }
}

```

```

coverage[j,k] <- (1-cont)*100
Vies[j,k]      <- mean(Analise[,1])-true[k]
EQM[j,k]       <- var(Analise[,1]) + Vies[j,k]^2
}
}

namesParaprec   <- c("precs")
for(k in 1:1){
  for(j in c(4:9,16:21,28:33)){
    if(j>=4&j<=9){ind <- j-3
  }else if(j>=16&j<=21){ind <- j-9
  }else if(j>=28&j<=33) ind <- j-15

Analise <- read.table(paste(get(noquote(namesParaprec)[k]),
".txt",sep="")[ind],h=TRUE)
if(ncol(Analise)==5) Analise <- Analise[,-c(2,4)]
colnames(Analise) <- c("mean","low","high")
head(Analise)
if(j==1||j==2||j==3|| j==13||j==14||j==15|| j==25||j==26||j==27){
true <- par_binomial
}else if(j==4||j==5||j==6|| j==16||j==17||j==18|| j==28||j==29||j==30){
true <- par_beta
}else if(j==7||j==8||j==9|| j==19||j==20||j==21|| j==31||j==32||j==33){
true <- par_gamma
}else if(j==10||j==11||j==12|| j==22||j==23||j==24|| j==34||j==35||j==36){
true <- par_poisson
}
xA=matrix(NA,ncol=1,nrow=nrow(Analise))
for(i in 1:nrow(Analise)){
if((Analise[i,2]>true[k+2] || Analise[i,3]<true[k+2]) == "TRUE") xA[i,]<-1 else xA[i,]<-0
}
cont = apply(xA,2,sum)/(nrow(xA))
coverage[j,k+2] <- (1-cont)*100
Vies[j,k+2]      <- mean(Analise[,1])-true[k+2]
EQM[j,k+2]       <- var(Analise[,1]) + Vies[j,k+2]^2
}
}

coverage

```

```
abs(Vies)
```

```
EQM
```

```
outrosNomes <- c("DICs","CPOs","WAICs","Timess")
```

```
OUTROS      <- matrix(NA,ncol=length(outrosNomes),nrow=length(beta0s))
```

```
dimnames(OUTROS)[2] <- list(outrosNomes)
```

```
dimnames(OUTROS)[1] <-
```

```
  list(paste(rep(c("Binomial_", "Beta_", "Gama_", "Poisson_"), rep(3,4)),
```

```
  c("gaussian", "laplace", "simplified_laplace"),
```

```
  rep(c("_50", "_100", "_200"), rep(12,3)), sep=' '))
```

```
for(k in 1:length(outrosNomes)){
```

```
for(j in 1:length(DICs)){
```

```
OUTRO  <- read.table(paste(get(noquote(outrosNomes)[k]),
```

```
  ".txt", sep="") [j], h=TRUE)
```

```
OUTROS[j,k] <- mean(abs(OUTRO$V1), na.rm=TRUE)
```

```
}}
```

```
OUTROS
```