# Comparação de estratégias de geração de propostas no algoritmo Metropolis-Hastings para um modelo Poisson log-linear

**Estevão Batista do Prado**

Departamento de Estatística - ICEx - UFMG

Fevereiro de 2016

# Comparação de estratégias de geração de propostas no algoritmo Metropolis-Hastings para um modelo Poisson log-linear

**Estevão Batista do Prado**

Orientador: Vinícius Diniz Mayrink

Dissertação submetida ao Programa de Pós-Graduação em Estatística da Universidade Federal de Minas Gerais, como parte dos requisitos necessários à obtenção do grau de Mestre em Estatística.

Departamento de Estatística
Instituto de Ciências Exatas
Universidade Federal de Minas Gerais

Belo Horizonte, MG - Brasil
Fevereiro de 2016

À minha família.

# Agradecimentos

Primeiramente a Deus pela vida, saúde e pelas ricas oportunidades.

Ao meu orientador, professor Vinícius Diniz Mayrink, pela acessibilidade, paciência, pela constante e cuidadosa orientação e por todo o conhecimento transmitido.

Aos professores da pós-graduação em Estatística da UFMG com quem tive contato.

Aos colegas da pós-graduação pela amizade e por todas as proveitosas discussões acadêmicas, em especial a A. Rafaella, Fernanda, Juliana, L. Gustavo, L. Sayuri, Raquel, Roberto e Vitor.

À professora Carminha e aos professores Enrico Colosimo e Antônio Ribeiro por toda paciência, empenho e sensibilidade ao momento em que não tive bolsa de estudos; pela oportunidade de ter feito parte do grupo "*Statistics in Medicine*" e por todo conhecimento compartilhado.

Aos amigos do grupo "*Statistics in Medicine*": José Luiz, Rodrigo Reis e Paulo Cerqueira pela receptividade e pelo ambiente prazeroso e motivador de pesquisa.

Ao professor Marcelo Azevedo Costa, meu orientador de estágio, pela oportunidade e conhecimentos transmitidos ao longo do projeto. Mais do que um excelente orientador, o professor Marcelo é um profissional competente, dedicado e organizado que contribuiu enormemente para minha formação.

À minha família "mineira": tia Sônia, tia Julita, priminho Matheus, prima M. Eduarda, prima Simone e primo Tuco. Serei eternamente grato pelo amor, carinho, respeito e suporte recebidos durante todo o período que estive em Belo Horizonte.

À minha família "paranaense": mãe, irmãos, irmã, tios, tias, primos, primas e avós. Obrigado pelo apoio e torcida. Meu combustível durante esse longo período foram vocês.

À FAPEMIG, pelo apoio parcial durante o mestrado através de uma bolsa de estudos.

Aos participantes da banca examinadora.

# Resumo

Os métodos de Monte Carlo via Cadeias de Markov (MCMC) são uma classe de algoritmos de simulação que, no contexto de inferência Bayesiana, são comumente utilizados para gerar amostras de forma indireta de uma distribuição *a posteriori* da qual conhecemos apenas o núcleo. O algoritmo Metropolis-Hastings *Random Walk* é um algoritmo MCMC bastante utilizado no contexto Bayesiano, e que gera bons resultados de estimativas *a posteriori* se a matriz de covariâncias da distribuição de propostas é bem especificada. Em situações de alta dimensão, a escolha dessa matriz não é trivial. Este trabalho tem como objetivo principal comparar diferentes estratégias com relação a geração de valores candidatos no Metropolis-Hastings que se diferem, basicamente, pela especificação da matriz de covariâncias da distribuição de propostas. Algoritmos adaptativos e não-adaptativos serão considerados. A comparação dos algoritmos é feita em cenário de simulação e em uma análise de dados reais com o modelo Poisson log-linear em um problema para dados de contagem com estrutura longitudinal. Os critérios utilizados para avaliar a *performance* dos métodos foram: o tamanho efetivo da amostra, que é uma função da correlação das cadeia dos parâmetros, e a precisão das estimativas pontuais e intervalares *a posteriori*. De forma geral, os resultados numéricos mostram que os algoritmos estimam bem os parâmetros de interesse e se diferenciam quanto ao *mixing* das cadeias e ao tempo computacional. Destaque para as opções adaptativas *Adaptive Metropolis*, *Robust Adaptive Metropolis* e *Iterative Weighted Least Squares Metropolis*.

Palavras-chave: inferência Bayesiana, métodos MCMC, Metropolis adaptativo, Simulação.

# Abstract

The Markov Chain Monte Carlo methods (MCMC) are a class of simulation algorithms widely used in Bayesian inference to indirectly draw samples from the posterior distribution, which is known up to a constant of proportionality. The random walk Metropolis-Hastings algorithm is a popular case providing good posterior estimates if the covariance matrix of the proposal distribution is well specified. In high dimensional situations, the specification of this matrix is not trivial. This dissertation aims to carry out comparisons between different strategies to generate candidates through Metropolis-Hastings algorithms, that basically differ in terms of the choice of covariance matrix of the proposal distribution. Adaptive and non-adaptive algorithms are considered. The comparison is made through a simulation study and an analysis of real data set using a Poisson log-linear model with longitudinal count structure. The criteria used to evaluate the performance of the algorithms are: the effective sample size, which is a function of the chain's autocorrelation, and the accuracy of the posterior point and interval estimates. In general, numerical results show that the algorithms estimate well the parameters of interest and they differ with respect to the mixing of the chains and to the computational time, especially the adaptive cases: Adaptive Metropolis, Robust Adaptive Metropolis and Iterative Weighted Least Squares Metropolis.

Keywords: Bayesian inference, Markov Chain Monte Carlo methods, Adaptive Metropolis, Simulation.

# Lista de abreviações

As siglas de abreviações estão em inglês, porém os significados estão em português.

| | |
|---|---|
| AM | Metropolis adaptativo |
| AM-HA | Metropolis adaptativo, versão de Haario et al. (2001) |
| AM-RR | Metropolis adaptativo, versão de Roberts e Rosenthal (2009) |
| AP | Metropolis proposta adaptativa |
| ARS | Método da rejeição adaptativo |
| ARMS | Método da rejeição adaptativo com Metropolis |
| DIC | Critério de informação da Deviance |
| GLM | Modelos Lineares Generalizados |
| HPD | Maior densidade a *a posteriori* |
| IWLS | Mínimos Quadrados Ponderados Iterativos |
| LPML | Critério log-pseudo verossimilhança marginal |
| MCMC | Método de Monte Carlo via Cadeias de Markov |
| p.d.f | Função densidade de probabilidade |
| PGM | Mistura Gaussiana Penalizada |
| p.m.f | Função massa de probabilidade |
| RS | Método da rejeição |
| RAM | Metropolis adaptativo Robusto |
| RWM | Metropolis passeio aleatório |
| SS | *Slice Sampling* |
| VBAM | Metropolis Adaptativo Variacional Bayesiano |
| VB-AKF | Filtro de Kalman Adaptativo Variacional Bayesiano |

WAIC Critério de informação de Watanabe-Akaike

# Lista de Símbolos

### Notações gerais

| | |
|---|---|
| $a$, $b$, $c$, $A$, $B$, $C$... | Escalares |
| $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{x}_i$, $\boldsymbol{\alpha}_i$, $\boldsymbol{\alpha}$... | Vetores |
| $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\boldsymbol{\Gamma}$... | Matrizes |
| $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$... | Conjuntos |
| $\mathbb{A}$, $\mathbb{B}$, $\mathbb{C}$, $\Theta$... | Espaços paramétricos |

### Símbolos específicos

| | |
|---|---|
| $\mathbf{A}_t$, $\mathbf{P}_t^-$, $\boldsymbol{\Sigma}_t$ | Matrizes na $t$-ésima iteração |
| $\mathbf{A}_t^\top$ | Transposta de $\mathbf{A}_t$ |
| $\mathbf{A}_t^{-1}$ | Inversa de $\mathbf{A}_t$ |
| $b()$, $c()$, $g()$, $t()$ | Funções |
| $b'()$ | Primeira derivada da função $b()$ |
| $b''()$ | Segunda derivada da função $b()$ |
| $d_1$, $d_2$, $g_1$, $g_2$ | Parâmetros da distribuição Gama Inversa |
| $\mathbb{E}_f[.]$ | Esperança de . com respeito a $f()$ |
| $f()$ | Distribuição de probabilidade |
| $g^{-1}()$ | Inversa da função $g()$ |
| $h(\theta_k)$ | logaritmo de $p(\theta_k\|\boldsymbol{\theta}_{-k})$ |
| $\mathbf{I}_d$ | Matriz identidade de dimensão $d$ |
| $\mathbf{K}_t^{(j)}$ | Matriz na $t$-ésima iteração de um algoritmo e na $j$-ésima de outro |
| $\mathcal{L}(\boldsymbol{\theta}\|\mathbf{a})$ | Função de verossimilhança associada à quantidade $\boldsymbol{\theta}$ |

| | |
|---|---|
| $m_r(\theta_k)$ | Função do ARS |
| $m'_r(\theta_k)$ | Função do ARMS |
| $\mathbf{m}_t$, $\mathbf{m}_t^-$, $\mathbf{x}_t$, $\ddot{\mathbf{y}}_t$, $\boldsymbol{\beta}^{(t)}$, $\boldsymbol{\theta}^{(t)}$ | Vetores na $t$-ésima iteração |
| $n_{ef}$ | Estatística tamanho efetivo da amostra |
| $N()$ | Distribuição Normal (ou Gaussiana) |
| $N_d()$ | Distribuição Normal multivariada $d$-dimensional |
| $q()$ | Distribuição de propostas |
| $v_t$, $v_t^-$, $\lambda_t$ | Escalares na $t$-ésima iteração |
| $\mathbf{W}(\boldsymbol{\beta})$ | Matriz de pesos de dimensão $N \times d$ |
| $w_i$ | Componente da linha e coluna $i$ da matriz $\mathbf{W}(\boldsymbol{\beta})$ |
| $\tilde{\mathbf{y}}(\boldsymbol{\beta})$ | Vetor coluna $N$-dimensional de observações transformadas |
| $\tilde{y_i}$ | Componente $i$ do vetor $\tilde{\mathbf{y}}(\boldsymbol{\beta})$ |
| $\boldsymbol{\beta}^*$, $\boldsymbol{\theta}^*$ | Vetores de valores candidatos |
| $\boldsymbol{\gamma}_{-i}$, $\boldsymbol{\delta}_{-iv}$ | Vetores sem o componente indicado no subíndice |
| $\theta_k^{(t)}$ | K-ésima entrada do vetor $\boldsymbol{\theta}$ na $t$-ésima iteração Monte Carlo |
| $\rho_l$ | Coeficiente de autocorrelação de espaçamento (ou $lag$) $l$ |
| $\sigma_\gamma^2$, $\sigma_\delta^2$ | Variância dos efeitos aleatórios |
| $\tau_t$ | Taxa de aceitação na $t$-ésima iteração |
| $\tau_*$ | Taxa de aceitação desejada |

# Sumário

# Capítulo 1

# Introdução

Os métodos de Monte Carlo via Cadeias de Markov (MCMC) são uma classe de algoritmos de simulação que fornecem boas aproximações para complicadas integrações, especialmente de alta dimensão (Gamerman e Lopes, 2006; Robert e Casella, 2009, 2004). No contexto de inferência Bayesiana, a escolha da distribuição de propostas no Metropolis-Hastings é um importante aspecto a ser considerado objetivando obter estimativas razoáveis *a posteriori*. No Metropolis-Hastings *Random Walk* (Metropolis et al., 1953; Hastings, 1970), por exemplo, é necessário a especificação dos parâmetros de locação e variabilidade da distribuição Normal geradora de propostas. O parâmetro de locação é centrado no valor da estimativa atual (da quantidade de interesse) e o de variabilidade é fixado *ad hoc*. Em diversas situações é difícil escolher um parâmetro de variabilidade que permita a distribuição de propostas gerar valores candidatos ao longo do espaço paramétrico da quantidade de interesse (Roberts et al., 1997; Haario et al., 1999). Além disso, uma escolha inapropriada pode levar a uma baixa taxa de aceitação, que por sua vez pode acarretar uma convergência lenta (ou até mesmo a não convergência) para distribuição alvo do parâmetro de interesse.

Diferentes estratégias foram propostas para lidar com esse problema em diversos contextos. Gamerman (1997) introduziu uma metodologia Bayesiana para fazer a estimação dos parâmetros em Modelos Lineares Generalizados Mistos (*Generalized Linear Mixed Models* - GLMM), generalizando a versão Bayesiana do algoritmo Mínimos Quadrados Ponderados Iterativos (*Weighted Least Squares* - WLS) proposto por West (1985). A

versão Bayesiana do WLS foi desenvolvida para a função de ligação identidade em Modelos Lineares Generalizados (*Generalized Linear Models* - GLM). A extensão para as demais funções de ligação é direta e Gamerman (1997) utilizou a ideia ampliando-a no contexto de GLMM.

Haario et al. (2001) propuseram um outro método bastante interessante que melhora os resultados do algoritmo Metropolis-Hastings. De acordo com os autores, um algoritmo adaptativo pode ser uma alternativa para lidar com a questão da especificação da distribuição de propostas. O algoritmo Metropolis Adaptativo (*Adaptive Metropolis* - AM), proposto pelos autores, utiliza todo o histórico do processo para "tunar" a distribuição de propostas. O AM é similar ao algoritmo Metropolis *Random Walk* (*Random Walk Metropolis* - RWM), com uma única diferença: a cada iteração ele calcula a covariância dos estados da cadeia e a utiliza para configurar a matriz de covariâncias da distribuição de propostas.

Em um trabalho recente, Vihola (2012) introduziu o algoritmo Metropolis Adaptativo Robusto (*Robust Adaptive Metropolis* - RAM), o qual possui uma regra que permite atualizar a matriz de covariâncias da distribuição de propostas ao mesmo tempo que se mantém a média da taxa de aceitação em um nível predeterminado. Uma restrição que o algoritmo impõe é que a distribuição geradora de propostas precisa ser Normal ou t-Student. O autor também mostra, através de resultados empíricos, que o desempenho do algoritmo RAM, quando a distribuição alvo do parâmetro de interesse não possui segundo momento finito, é promissor quando comparado a outros algoritmos adaptativos como o AM e o Metropolis Escalar Adaptativo dentro do AM (Atchadé e Fort, 2010; Andrieu e Thoms, 2008).

Mbalawata et al. (2015) também apresentaram um novo algoritmo Metropolis adaptativo. A ideia, de uma forma bem geral, é usar o Filtro de Kalman Adaptativo Variacional Bayesiano (*Variational Bayesian Adaptive Kalman Filter* - VB-AKF), proposto por Särkkä e Nummenmaa (2009) e Särkkä e Hartikainen (2013), para estimar a matriz de covariâncias, sendo a distribuição de propostas Gaussiana. Para provar a convergência do algoritmo são utilizados conceitos de Filtro de Kalman e a Lei Forte dos Grandes Números.

O principal objetivo do presente trabalho é comparar o desempenho de algoritmos Metropolis-Hastings que se diferenciam quanto a geração de valores candidatos. Os métodos descritos por Gamerman (1997), Haario et al. (2001), Vihola (2012) e Mbalawata et al. (2015) serão avaliados em simulações replicando dados reais com estrutura longitudinal de contagem. O algoritmo RWM também é estudado.

Todos os algoritmos que iremos investigar neste trabalho foram implementados no *software* R (R Core Team, 2015). O R é um *software* livre e bastante popular como ferramenta computacional estatística. Ele pode ser executado nos principais sistemas operacionais e conta com pacotes auxiliares que são constantemente criados e atualizados pelos usuários.

## 1.1 Organização da dissertação

O Capítulo 2, apresentado a seguir, irá descrever conceitos básicos de inferência Bayesiana. Esses conceitos são necessários para um melhor entendimento dos próximos capítulos para aqueles leitores que não possuem familiaridade com a abordagem Bayesiana.

O Capítulo 3 descreve os métodos MCMC, os diversos algoritmos Metropolis-Hastings, que serão utilizados nesse trabalho, bem como o critério considerado para comparação dos métodos.

O Capítulo 4 introduz o modelo Poisson log-linear para dados longitudinais de contagem, abordando a função de verossimilhança e as distribuições condicionais completas e a necessidade do uso do algoritmo Metropolis-Hastings. Esse modelo faz parte da classe GLM e é apropriado para todas as metodologias que serão utilizadas nesse trabalho. Também é feita a descrição dos dados que serão considerados no estudo de simulação e na aplicação real.

O Capítulo 5 apresenta um estudo de simulação em que os dados têm estrutura longitudinal. Nesse capítulo, detalhamos como foi feita a simulação dos dados. O objetivo é avaliar a *performance* dos métodos em termos de resultados de inferência em um cenário no qual os valores reais dos parâmetros são conhecidos.

O Capítulo 6 desenvolve uma comparação de algoritmos usando dados reais de um estudo longitudinal com pacientes que sofrem de ataques epilépticos. Adicionalmente, são realizadas comparações de resultados assumindo diferentes distribuições para os efeitos aleatórios do modelo Poisson log-linear.

O Capítulo 7 fará um resumo de tudo aquilo que foi apresentado e discutido nesta dissertação. Destaque será dado às principais conclusões tiradas nos estudos. O encerramento é feito com algumas propostas de trabalhos futuros.

# Capítulo 2

# Conceitos básicos de inferência Bayesiana

No contexto de inferência sob o paradigma Bayesiano, objetivamos, a partir de um conjunto de informações númericas e subjetivas, tirar conclusões sobre um determinado fenômeno de interesse utilizando modelos probabilísticos. Sob a ótica frequentista, também usa-se modelos probabilísticos, mas a informação *a priori* que pode-se ter sobre a quantidade de interesse não é incorporada ao processo de decisão. Isto é, na inferência Clássica apenas os dados são utilizados no processo decisório enquanto que na inferência Bayesiana os dados e toda informação *a priori* que se tem sobre o fenômeno de interesse são considerados.

Seja $\boldsymbol{\theta} \in \Theta$ uma quantidade desconhecida, a qual temos interesse em estudar, e $\Theta$ o espaço paramétrico de $\boldsymbol{\theta}$, dado uma amostra independente e identicamente distribuída (i.i.d.) $\mathbf{y} = (y_1, y_2, ..., y_N)$ da distribuição $p$ indexada por $\boldsymbol{\theta}$, a função de verossimilhança associada à quantidade $\boldsymbol{\theta}$ é

$$\mathcal{L}(\boldsymbol{\theta}|\mathbf{y}) = \prod_{i=1}^{N} p(y_i|\boldsymbol{\theta}). \tag{2.1}$$

Em essência, toda informação fornecida por cada $y_i$ sobre $\boldsymbol{\theta}$ está contida na função de verossimilhança. A partir disso, a estimativa para $\boldsymbol{\theta}$ pelo método de Máxima Verossimilhança é a menor das cotas superiores que maximiza a função dada em (2.1). Em paralelo, a estimativa para $\boldsymbol{\theta}$ via inferência Bayesiana se dá através da distribuição *a*

*posteriori* definida pela regra de Bayes da seguinte forma

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{\mathcal{L}(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta})}{\int_{\Theta} \mathcal{L}(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta})d\boldsymbol{\theta}}, \tag{2.2}$$

que também pode ser reescrita como segue abaixo, pois a quantidade no denominador em (2.2) não depende de $\boldsymbol{\theta}$ e, nesse caso, é considerada uma constante;

$$p(\boldsymbol{\theta}|\mathbf{y}) \propto \mathcal{L}(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta}). \tag{2.3}$$

Neste caso, $\boldsymbol{\theta}$ possui uma distribuição de probabilidade que pode ser usada na obtenção de estimativas pontuais e intervalares. O componente $p(\boldsymbol{\theta})$ é chamado de distribuição *a priori*. A ideia por trás da inserção de $p(\boldsymbol{\theta})$ é de que todo o conhecimento inicial disponível sobre $\boldsymbol{\theta}$ deve ser considerado na análise. Isto é, $p(\boldsymbol{\theta})$ resume tudo o que se sabe sobre a quantidade de interesse antes de observarmos os dados. A escolha de $p(\boldsymbol{\theta})$ é um importante aspecto a ser considerado, uma vez que esse componente influencia os resultados de inferência. No entanto, a medida que o número de observações $N$ cresce, a influência da distribuição *a priori* sobre os resultados de inferência diminui gradativamente. Em diversas situações é difícil escolher uma distribuição de probabilidade que traduz todo o conhecimento sobre a quantidade de interesse, pois não se tem muito conhecimento sobre $\boldsymbol{\theta}$ ou a informação que se tem não é confiável. Nessas situações, as distribuições *a priori* não-informativas surgem como uma classe de distribuições que têm como objetivo principal minimizar o impacto da escolha de $p(\boldsymbol{\theta})$ nos resultados de inferência (Marin e Robert, 2007).

Existem diversos tipos de distribuições *a priori* não-informativa. Uma distribuição *a priori* para $\boldsymbol{\theta}$ é dita não-informativa, no sentido Bayes-Laplace, quando assume-se uma distribuição uniforme em $\Theta$. Isto é, uma vez que não se tem informação sobre o parâmetro de interesse, atribui-se mesma probabilidade para todos os valores do seu espaço paramétrico. Em paralelo, $p(\boldsymbol{\theta})$ é dita não-informativa, no sentido Jeffreys, quando é proporcional a raiz quadrada da Informação de Fisher de $\boldsymbol{\theta}$ . Em ambos os sentidos, pode-se ter distribuição *a priori* imprópria, isto é, distribuição que não integra 1, no entanto, ela acaba sendo usada pois em alguns casos a distribuição *a posteriori* é própria (Gelman et al., 2014). Neste trabalho, utilizaremos distribuições *a priori* que carregam

certa informação sobre o parâmetro de interesse ao mesmo tempo que apresentam grande variabilidade sobre essa informação. Esse tipo de distribuição *a priori* é dita pouco informativa ou distribuição vaga.

Uma vez que a distribuição *a posteriori* foi obtida é possível determinar outras distribuições que podem ser de interesse, como a distribuição preditiva *a posteriori*, que é usada para predizer valores ainda não observados $\tilde{\mathbf{y}}$, e a distribuição marginal de $\mathbf{y}$.

$$p(\tilde{\mathbf{y}}|\mathbf{y}) = \int_{\Theta} p(\tilde{\mathbf{y}}|\boldsymbol{\theta}, \mathbf{y}) p(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta}, \qquad \tilde{\mathbf{y}} \sim p(\tilde{\mathbf{y}}|\boldsymbol{\theta}, \mathbf{y}),$$
$$p(\mathbf{y}) = \int_{\Theta} \mathcal{L}(\boldsymbol{\theta}|\mathbf{y}) p(\boldsymbol{\theta}) d\boldsymbol{\theta},$$

É importante ressaltar que a constante que foi desconsiderada em (2.3) se faz necessária tanto no cálculo da distribuição preditiva *a posteriori* quanto no cálculo da distribuição marginal de $\mathbf{y}$. Não é difícil encontrar problemas onde a distribuição *a posteriori* não possui forma fechada e/ou não pode ser tratada analiticamente. Para essas situações, existem métodos computacionais que fornecem aproximações bastante boas. Destaque para os métodos MCMC, que serão abordados neste trabalho através dos algoritmos Metropolis-Hastings e *Gibbs Sampling*.

## 2.1 Análise conjugada

Em algumas situações, as distribuições *a priori* podem ser escolhidas de forma a facilitar, principalmente em termos analíticos, a obtenção da distribuição *a posteriori*. As distribuições conjugadas *a priori* são uma classe de distribuições proporcionais ao núcleo de uma dada função de verossimilhança de modo que a distribuição *a posteriori* está nessa mesma classe (Robert, 2007). Isto é, dizemos que $\mathcal{C} = \{p(\boldsymbol{\theta}|\mathbf{h}); h \in \mathbb{H}\}$, onde $\mathbb{H}$ representa o espaço paramétrico de hiperparâmetros, é uma família de distribuições conjugadas para $\mathcal{F} = \{p(\mathbf{y}|\boldsymbol{\theta}); \boldsymbol{\theta} \in \Theta)\}$ se, para todo $p(\boldsymbol{\theta}|\mathbf{h}) \in \mathcal{C}$ e $p(\mathbf{y}|\boldsymbol{\theta}) \in \mathcal{F}$, a distribuição *a posteriori* $p(\boldsymbol{\theta}|\mathbf{y}) \in \mathcal{C}$.

No contexto deste trabalho, o modelo Poisson log-linear, que será apresentado no Capítulo 4, não possui distribuição conjunta *a posteriori* que pode ser tratada analiticamente. No contexto do *Gibbs Sampling*, não foi possível escolher distribuições *a priori*

que permitissem uma análise conjugada para alguns dos parâmetros do modelo. A conjugação foi possível apenas para obter a condicional completa *a posteriori* das variâncias dos efeitos aleatórios incluídos na modelagem; estes detalhes sobre o modelo Poisson log-linear serão apresentados no Capítulo 4.

## 2.2 Conclusões do capítulo

Neste capítulo apresentamos alguns conceitos básicos de inferência Bayesiana no intuito de expor o paradigma Bayesiano, apontando diferenças conceituais entre as inferências Clássica e Bayesiana. Em termos gerais, tomar decisão sob o paradigma da inferência Clássica é basear-se unicamente nos dados em questão. Por outro lado, sob a ótica Bayesiana, o processo decisório é construído através da combinação "dados" mais "informação *a priori*". A informação *a priori* carrega todo o conhecimento prévio que se tem sobre o problema através de uma distribuição de probabilidade. Em situações onde se tem pouco ou nenhuma informação, é possível escolher distribuições *a priori* não-informativas, que visam minimizar o impacto da escolha da distribuição *a priori* na tomada de decisão. A partir disso, as conclusões sobre o problema em questão são extraídas da distribuição *a posteriori*, que por vezes pode não ser obtida analiticamente e métodos computacionais se fazem necessários.

# Capítulo 3

# Métodos MCMC

Neste capítulo apresentaremos o método de integração de Monte Carlo, uma breve revisão de teoria de cadeias de Markov, os métodos que serão utilizados nas simulações e análise do banco de dados com informações de pacientes epilépticos e o critério que será utilizado na comparação dos algoritmos.

## 3.1 Integração de Monte Carlo

No contexto de inferência Bayesiana, problemas envolvendo integrais de distribuições de probabilidade são frequentemente encontrados. Isto é, em algumas situações não é possível avaliar analiticamente a distribuição conjunta *a posteriori* e, a partir da mesma, extrair as distribuições marginais de interesse. Isso se dá, principalmente, pela dificuldade no cálculo de integrais de alta dimensionalidade e/ou simplesmente pela dificuldade do cálculo (Robert e Casella, 2009). Nesse sentido, o método de integração de Monte Carlo é uma ferramenta útil e interessante. De forma genérica, esse método fornece uma aproximação bastante razoável para integrações do tipo

$$\mathbb{E}_f[t(\theta)] = \int_{\Theta} t(\theta) f(\theta) d\theta, \tag{3.1}$$

sendo $\Theta$ o conjunto onde a variável $\theta$ toma valores; geralmente $\Theta$ é igual ao suporte da p.d.f $f(.)$. Em termos gerais, para fornecer uma aproximação para (3.1) o método de Monte Carlo gera amostras $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(K)}$ (aleatórias e i.i.d's.) de $f(.)$ e as aplica na

função $t(.)$ e, posteriormente, toma a média dos valores resultantes da seguinte forma

$$\bar{t} = \frac{1}{K} \sum_{t=1}^{K} t(\theta^{(t)}). \tag{3.2}$$

Para um número de amostras $K$ suficientemente grande, a Lei Forte dos Grandes Números garante que (3.2) converge quase certamente para $\mathbb{E}_f[t(\theta)]$, sendo $t(.)$ uma função integrável (Robert e Casella, 2004).

## 3.2 Alguns conceitos de cadeias de Markov

Antes de descrever os algoritmos MCMC, uma breve introdução à cadeia de Markov (em tempo discreto) será feita com o intuito de deixar mais clara algumas notações que serão utilizadas adiante.

Uma cadeia de Markov $\{\theta^{(t)}\}_{t\in\mathbb{N}}$ é uma sequência de variáveis aleatórias dependentes de modo que a distribuição de $\theta^{(t)}$, dado todas as variáveis aleatórias que a antecederam, depende somente de $\theta^{(t-1)}$. A distribuição condicional de $\theta^{(t)}$ dado $\theta^{(t-1)}$ é chamada de núcleo de transição ou distribuição de transição.

$$\left(\theta^{(t)}|\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, ..., \theta^{(t-1)}\right) \sim q(\theta^{(t)}, \theta^{(t-1)}).$$

Dizer que uma cadeia de Markov $\{\theta^{(t)}\}_{t\in\mathbb{N}}$ possui distribuição de probabilidade estacionária implica em dizer que existe uma função $f(.)$ tal que ela é a p.d.f ou p.m.f de $\theta^{(t)}$ e $\theta^{(t-1)}$. Além disso, se uma cadeia de Markov possui distribuição estacionária, então ela é irredutível, recorrente e aperiódica (Norris (1998), Teorema 1.8.3). A irredutibilidade está ligada ao fato da distribuição de transição permitir, independente do ponto inicial, que a cadeia se mova ao longo do conjunto onde $\theta^{(t)}$ toma valores, chamado de espaço de estados, com probabilidade positiva. Por sua vez, a recorrência está relacionada ao número esperado de retornos da cadeia a um ponto qualquer do seu espaço de estado. Se, para todos os pontos no espaço de estados, o número esperado de retornos é infinito, então a mesma é recorrente, caso contrário, é chamada de transiente. Adicionalmente, um estado $\theta^{(t)}$ de uma cadeia de Markov é periódico se a probabilidade da cadeia retornar a ele somente nos tempos $m$, $2m$, $3m$, ..., com $m > 1$, for maior que zero; caso

contrário, ele é dito aperiódico. A recorrência e a aperiodicidade podem ser propriedades da cadeia ou de alguns estados. Por exemplo, no caso da cadeia ser irredutível, a recorrência e a aperiodicidade serão propriedades da cadeia. No caso de a cadeia não ser irredutível, estas podem ser propriedades de alguns estados. Os métodos MCMC gozam das propriedades mencionadas acima, isto é, geram cadeias de Markov irredutíveis, recorrentes, aperiódicas e que, por construção, possuem distribuição estacionária (Robert e Casella, 2004). A partir disso, o teorema ergódico garante, para um número de amostras $T$ suficientemente grande, que

$$\bar{h} = \frac{1}{T}\sum_{t=1}^{T} t(\theta^{(t)}) \xrightarrow{\text{q.c}} \mathbb{E}_f[t(x)],$$

sendo que "$\xrightarrow{\text{q.c}}$" indica convergência quase certa e $\theta^{(t)}$'s amostras da p.d.f ou p.m.f $f(.)$, as quais apresentam uma relação de dependência Markoviana, e $t(.)$ uma função integrável. Para um tratamento mais completo sobre cadeias de Markov veja Norris (1998) e Feller (1967).

## 3.3 *Gibbs Sampling*

O algoritmo *Gibbs Sampling* foi o primeiro método MCMC a ser largamente utilizado em inferência Bayesiana. Inicialmente proposto por Geman e Geman (1984), no contexto de processamento de imagem, e posteriormente difundido na comunidade estatística por Gelfand e Smith (1990), o *Gibbs Sampling* é um método de simulação estocástica via cadeias de Markov que permite amostrar de uma determinada densidade de interesse a partir de suas distribuições condicionais completas.

No intuito de obter uma amostra da distribuição conjunta *a posteriori*, o algoritmo *Gibbs Sampling* surge como uma alternativa bastante interessante que permite amostrar da conjunta a partir das distribuições condicionais completas, quando estas apresentam forma fechada. Existem situações em que nem todas as condicionais completas são conhecidas. Nestes casos, conforme veremos mais adiante, podemos ainda usar a estrutura do *Gibbs Sampling* com passos de outros algoritmos; dentre eles podemos citar o Metropolis-Hastings, o *Adaptive Rejection Sampling* - ARS (Gilks, 1992; Gilks e Wild,

1992), o *Adaptive Rejection Metropolis Sampling* - ARMS (Gilks et al., 1995), o *Slice Sampling* (Neal, 2003)[1], entre outros.

Suponha que $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_d)$ é um vetor de parâmetros de interesse e que queremos obter uma amostra da distribuição conjunta *a posteriori* de $\boldsymbol{\theta}$, a qual denotaremos por $p(\boldsymbol{\theta}|\mathbf{D})$, sendo $\mathbf{D}$ os dados (variável resposta e covariáveis). Em determinados problemas, $p(\boldsymbol{\theta}|\mathbf{D})$ não possui forma fechada e/ou possui alta dimensão. A partir disso, defini-se um conjunto de $d$ distribuições condicionais completas dadas da seguinte forma

$$p(\theta_k|\theta_1, ..., \theta_{k-1}, \theta_{k+1}, ..., \theta_d, \mathbf{D}), \text{ com } k = 1, 2, ..., d.$$

Com isso, utilizando o *Gibbs Sampling* obtemos uma amostra da distribuição conjunta *a posteriori* a partir de sucessivas amostragens das distribuições condicionais completas de cada $\theta_k$. O *Gibbs Sampling* pode ser descrito como segue.

---

**Algoritmo** *Gibbs Sampling*

1. Inicie $t = 1$ e defina valores iniciais $\theta_1^{(0)}, \theta_2^{(0)}, ..., \theta_d^{(0)}$ para o vetor $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_d)$.

2. Amostre

$$\theta_1^{(t)} \sim p(\theta_1|\theta_2^{(t-1)}, \theta_3^{(t-1)}, \theta_4^{(t-1)}, ..., \theta_d^{(t-1)}, \mathbf{D});$$

$$\theta_2^{(t)} \sim p(\theta_2|\theta_1^{(t)}, \theta_3^{(t-1)}, \theta_4^{(t-1)}, ..., \theta_d^{(t-1)}, \mathbf{D});$$

$$\theta_3^{(t)} \sim p(\theta_3|\theta_1^{(t)}, \theta_2^{(t)}, \theta_4^{(t-1)}, ..., \theta_d^{(t-1)}, \mathbf{D});$$

$$\vdots$$

$$\theta_d^{(t)} \sim p(\theta_d|\theta_1^{(t)}, \theta_2^{(t)}, \theta_3^{(t)}, ..., \theta_{d-1}^{(t)}, \mathbf{D});$$

3. Tome $t = t + 1$ e volte ao passo 2 até obter a amostra desejada após a convergência da cadeia para todos os $\theta_k$.

---

A ideia do *Gibbs Sampling* é obter uma amostra de $p(\boldsymbol{\theta}|\mathbf{D})$ a partir das distribuições condicionais completas *a posteriori* simulando (individualmente ou em bloco) cada componente do vetor $\boldsymbol{\theta}$ através de uma estrutura Markoviana. Sob certas condições de

[1] veja o Apêndice A para mais detalhes sobre esse algoritmo.

regularidade, tais como a existência de estacionariedade, irredutibilidade, recorrência e aperiodicidade da cadeia gerada para cada $\theta_k$, a amostra obtida a partir do algoritmo converge para uma amostra da distribuição conjunta *a posteriori*. Os Capítulos 9 e 10 de Robert e Casella (2004) fornecem definições e demonstrações sobre a convergência do *Gibbs Sampling*.

## 3.4 Metropolis-Hastings

Ainda no contexto de obter uma amostra da distribuição conjunta *a posteriori*, o algoritmo Metropolis-Hastings, proposto inicialmente por Metropolis et al. (1953), e posteriormente generalizado por Hastings (1970), simula valores de uma expressão para a qual não reconhecemos a distribuição. Uma diferença importante entre os algoritmos *Gibbs Sampling* e Metropolis-Hastings é que o último possui um passo de aceitação/rejeição de valores candidatos. Se todas as condicionais completas apresentam forma fechada então não haverá esse passo de aceitação/rejeição no *Gibbs Sampling*.

Suponha que queremos amostrar da distribuição conjunta *a posteriori* $p(\boldsymbol{\theta}|\mathbf{D})$. O Metropolis-Hastings é baseado em uma distribuição de transição $q(., \boldsymbol{\theta}^{(t-1)})$, a qual gera valores candidatos $\boldsymbol{\theta}^*$ sujeitos a uma probabilidade de aceitação, que indica a probabilidade deles serem aceitos como os próximos valores na sequência na cadeia. O algoritmo é descrito a seguir.

**Algoritmo** Metropolis-Hastings

1. Inicie $t = 1$ e defina valores iniciais $\theta_1^{(0)}, \theta_2^{(0)}, ..., \theta_d^{(0)}$ para o vetor $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_d)$;

2. Para $k = 1, 2, ..., d$

   (a) Amostre $\theta_k^*$ da distribuição geradora de propostas $q_k(., \theta_k^{(t-1)})$;

   (b) Calcule

   $$\alpha_k(\theta_k^{(t-1)}, \theta_k^*) = \min\left\{1, \frac{p(\theta_k^*|\boldsymbol{\theta}_{-k}^{(t-1)}, \mathbf{D}) q_k(\theta_k^{(t-1)}, \theta_k^*)}{p(\theta_k^{(t-1)}|\boldsymbol{\theta}_{-k}^{(t-1)}, \mathbf{D}) q_k(\theta_k^*, \theta_k^{(t-1)})}\right\}, \tag{3.3}$$

   sendo $\boldsymbol{\theta}_{-k}^{(t)}$ o vetor $\boldsymbol{\theta}$ sem o $k$-ésimo componente na $t$-ésima iteração e $p(\theta_k|\boldsymbol{\theta}_{-k}, \mathbf{D})$ o núcleo não reconhecido da distribuição alvo.

   (c) Gere $u_k \sim U[0, 1]$. Se $u_k < \alpha_k(\theta_k^{(t-1)}, \theta_k^*)$, faça $\theta^{(t)} = \theta^*$, caso contrário, faça $\theta_k^{(t)} = \theta_k^{(t-1)}$;

3. Tome $t = t + 1$ e retorne ao passo 2 até que a amostra desejada seja obtida após a convergência da cadeia para cada $\theta_k$.

Sob algumas condições de regularidade associadas a distribuição de propostas e a distribuição alvo, tais como irredutibilidade e estacionariedade, a amostra gerada pelo Metropolis-Hastings converge para uma amostra extraída da distribuição conjunta *a posteriori* $p(\boldsymbol{\theta}|\mathbf{D})$. No algoritmo descrito acima, cada componente do vetor $\boldsymbol{\theta}$ é gerado individualmente, porém vale ressaltar que também é possível gerar de $\boldsymbol{\theta}$ conjuntamente. A distribuição de propostas é geralmente escolhida de acordo com o espaço paramétrico dos componentes de $\boldsymbol{\theta}$. Por exemplo, se $\theta_k$ toma valores em $\mathbb{R}$ então é razoável escolher uma $q_k()$ de modo que ela também tome valores em $\mathbb{R}$.

Adicionalmente, se $q_k(\theta_k^*, \theta_k^{(t-1)})$ é uma distribuição simétrica em seus argumentos, isto é, $q_k(\theta_k^*, \theta_k^{(t-1)})$ é igual a $q_k(\theta_k^{(t-1)}, \theta_k^*)$ então (3.3) pode ser escrita da seguinte forma

$$\alpha_k(\theta_k^{(t-1)}, \theta_k^*) = \min\left\{1, \frac{p(\theta_k^*|\boldsymbol{\theta}_{-k}^{(t-1)}, \mathbf{D})}{p(\theta_k^{(t-1)}|\boldsymbol{\theta}_{-k}^{(t-1)}, \mathbf{D})}\right\}.$$

Neste caso, o Metropolis-Hastings é conhecido apenas como algoritmo Metropolis. O

Metropolis é chamado de Metropolis *Random Walk* (RWM) quando a distribuição de transição $q_k(.,\theta_k^{(t-1)})$ é a Normal, a qual é claramente simétrica em seus argumentos. No caso univariado $q_k(\theta_k^*,\theta_k^{(t-1)}) = (2\pi\omega)^{-\frac{1}{2}}\exp\{-\frac{1}{2\omega}(\theta_k^* - \theta_k^{(t-1)})^2\}$, sendo $\omega$ uma constante maior que zero.

Em alguns problemas é comum combinar os algoritmos *Gibbs Sampling* e Metropolis-Hastings, pois algumas distribuições condicionais completas apresentam forma fechada e outras não. Para as que possuem forma fechada, a amostragem é direta, para as que não possuem, o Metropolis-Hastings pode ser usado. Esse algoritmo "híbrido" é chamado de *Metropolis-within-Gibbs* e foi proposto por Muller (1991) e Muller (1993). Em sua versão original, dentro de uma única iteração do *Gibbs Sampling*, são feitas várias iterações até que a convergência seja atingida para todos os parâmetros que precisam do passo Metropolis-Hastings. No entanto, uma vez que a inclusão do passo Metropolis-Hastings não altera as propriedades de ergocidade verificadas para o *Gibbs Sampling*, verificou-se que é necessário apenas uma única iteração do Metropolis-Hastings; para mais detalhes veja Gamerman e Lopes (2006) e Robert e Casella (2004).

Além da versão RWM, o algoritmo Metropolis-Hastings possui outras versões. As versões *Independent* Metropolis (Mengersen e Tweedie, 1996) e ARMS, podem ser vistas como combinação do Metropolis-Hastings com o método *Rejection Sampling*. Em diversos problemas, esses algoritmos produzem bons resultados. No entanto, em situações de alta dimensão, o *Independent* Metropolis pode não ser uma alternativa interessante devido a complexidade da escolha da distribuição de propostas.

## 3.5 Metropolis IWLS

Gamerman (1997) apresentou uma metodologia Bayesiana para fazer estimação dos parâmetros em Modelos Lineares Generalizados Mistos (GLMM), a qual chamaremos de Metropolis via Mínimos Quadrados Ponderados Iterativos (IWLS) ou simplesmente Metropolis IWLS. De certa forma, o IWLS pode ser visto como uma generalização da versão Bayesiana do algoritmo de Mínimos Quadrados Ponderados (WLS) proposto por West (1985). Nesta seção, antes de apresentarmos o Metropolis IWLS, revisamos brevemente

Modelos Lineares Generalizados (GLM) e os algoritmos WLS clássico e Bayesiano.

No contexto de GLM, as informações analisadas consistem de $N$ indivíduos, para os quais são observados um conjunto de $d$ covariáveis e uma determinada resposta de interesse $Y_i$. As observações são assumidas independentes com p.d.f ou p.m.f na família exponencial se puderem ser escritas na forma

$$f(Y_i;\psi_i,\phi) = \exp\left\{\phi[Y_i\psi_i - b(\psi_i)] + c(Y_i;\phi)\right\}, \tag{3.4}$$

sendo o parâmetro de escala $\phi$ e as funções $b(.)$ e $c(.)$ conhecidas. Tem-se ainda que a média das respostas $Y_i$, $\mu_i = \mathbb{E}(Y_i|\psi_i) = b'(\psi_i)$ para $i = 1, 2, ..., N$, está relacionada às covariáveis por meio de uma função de ligação $g(.)$ contínua, estritamente monótona, duas vezes derivável e conhecida, a qual denotamos:

$$g(\mu_i) = \eta_i = \mathbf{x}_i\boldsymbol{\beta}, \tag{3.5}$$

com $\mathbf{x}_i = (1, x_{i1}, x_{i2}, ..., x_{id})$ representando o vetor linha de $d$ covariáveis do indivíduo $i$. Em situações de sobredispersão, isto é, em situações em que a variabilidade observada é maior do aquela predita pelo modelo, a expressão em (3.5) pode ser reescrita da seguinte maneira:

$$g(\mu_i) = \eta_i = \mathbf{x}_i\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{\alpha}_i, \quad \boldsymbol{\alpha}_i \sim N_q(\mathbf{0}, \mathbf{G}), \tag{3.6}$$

com $\boldsymbol{z}_i = (z_{i1}, z_{i2}, ..., z_{iq})$ representando o vetor linha de $q$ covariáveis adicionais associadas ao indivíduo $i$, $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, ..., \alpha_{iq})^\top$ o vetor dos efeitos aleatórios e $\mathbf{G}$ a matriz de covariâncias dos efeitos aleatórios. Geralmente as colunas de $\mathbf{z}_i$ são de 1's; essa representação é adotada por Gamerman (1997). A partir do momento em que adicionamos o componente aleatório ao preditor linear em (3.5), o modelo passa a se chamar GLMM. No Capítulo 4 serão definadas algumas notações para $\mathbf{X}$ e $\boldsymbol{\beta}$, no entanto, por uma questão de conveniência, apenas para essa seção, denote $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)$, $\mathbf{Z} = (\boldsymbol{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N)$, $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_d)^\top$ e $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_N)^\top$.

A estimação de $\boldsymbol{\beta}$ em (3.5) pode ser obtida através do algoritmo WLS. Para isso, define-se o vetor de observações transformadas $\tilde{\mathbf{y}}(\boldsymbol{\beta})$ e a matriz de pesos $\mathbf{W}(\boldsymbol{\beta})$ (McCullagh e Nelder, 1989). Dessa forma, o algoritmo pode ser descrito como segue abaixo.

1. Inicie $t = 1$ e $\boldsymbol{\beta} = \boldsymbol{\beta}^{(0)}$;

2. Calcule $\boldsymbol{\beta}^{(t)} = [\mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}^{(t-1)})\mathbf{X}]^{-1}\mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}^{(t-1)})\tilde{\mathbf{y}}(\boldsymbol{\beta}^{(t-1)})$;

3. Calcule $\mathbf{W}(\boldsymbol{\beta}^{(t)}) = [\mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}^{(t-1)})\mathbf{X}]^{-1}$;

4. Tome $t = t + 1$ e volte ao passo 2 até que a convergência seja atingida.

A matriz de pesos $\mathbf{W}(\boldsymbol{\beta})$ e o vetor de observações transformadas $\tilde{\mathbf{y}}(\boldsymbol{\beta})$ são dados por

$$\mathbf{W}(\boldsymbol{\beta}) = \text{diag}(w_1, w_2, ..., w_N) \text{ com } w_i^{-1} = \left(\frac{\partial \eta_i}{\partial \mu_i}\right)^2 \left(\frac{\partial \mu_i}{\partial \psi_i}\right) \equiv \{g'(\mu_i)\}^2 b''(\psi_i),$$

$$\tilde{\mathbf{y}}(\boldsymbol{\beta}) = (\tilde{Y_1}, \tilde{Y_2}, ..., \tilde{Y_N})^\top \text{ com } \tilde{Y_i} = \eta_i + (Y_i - \mu_i)\frac{\partial \eta_i}{\partial \mu_i} \equiv \eta_i + (Y_i - \mu_i)g'(\mu_i).$$

A versão Bayesiana do WLS foi desenvolvida por West (1985) para distribuições com função de ligação identidade, porém a extensão para as demais funções de ligação é direta. Assumindo que $\boldsymbol{\beta} \sim N_d(\mathbf{a}, \mathbf{C})$, o WLS Bayesiano pode ser descrito como segue abaixo:

1. Inicie $t = 1$ e $\boldsymbol{\beta} = \boldsymbol{\beta}^{(0)}$;

2. Calcule $\boldsymbol{\beta}^{(t)} = [\mathbf{C}^{-1} + \mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}^{(t-1)})\mathbf{X}]^{-1}[\mathbf{C}^{-1}\mathbf{a} + \mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}^{(t-1)})\tilde{\mathbf{y}}(\boldsymbol{\beta}^{(t-1)})]$;

3. Calcule $\mathbf{W}(\boldsymbol{\beta}^{(t)}) = [\mathbf{C}^{-1} + \mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}^{(t-1)})\mathbf{X}]^{-1}$;

4. Tome $t = t + 1$ e volte ao passo 2 até que a convergência seja atingida.

Note que o WLS Bayesiano é similar a sua versão clássica, com uma única diferença: a informação *a priori* sobre $\boldsymbol{\beta}$, assumindo que $\boldsymbol{\beta} \sim N_d(\mathbf{a}, \mathbf{C})$, é incorporada ao algoritmo através da matriz de precisão $\mathbf{C}^{-1}$ e do vetor $\mathbf{a}$. Se a distribuição *a priori* para $\boldsymbol{\beta}$ for vaga, no sentido de apresentar grande variabilidade, a matriz $\mathbf{C}^{-1}$ tende a uma matriz nula e a versão original do WLS é estabelecida.

O algoritmo Metropolis IWLS proposto por Gamerman (1997) pode ser visto como uma combinação da versão Bayesiana do WLS com o RWM. O IWLS é descrito a seguir.

**Algoritmo** Metropolis IWLS

1. Inicie $t = 1$ e $\boldsymbol{\beta} = \boldsymbol{\beta}^{(0)}$;

2. Amostre $\boldsymbol{\beta}^*$ da distribuição $N_d(\boldsymbol{\beta}^{(t-1)}, \mathbf{W}(\boldsymbol{\beta}^{(t-1)}))$;

   (a) $\boldsymbol{\beta}^{(t)} = [\mathbf{C}^{-1} + \mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}^{(t-1)})\mathbf{X}]^{-1}[\mathbf{C}^{-1}\mathbf{a} + \mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}^{(t-1)})\tilde{\mathbf{y}}(\boldsymbol{\beta}^{(t-1)})]$;

   (b) $\mathbf{W}(\boldsymbol{\beta}^{(t)}) = [\mathbf{C}^{-1} + \mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}^{(t-1)})\mathbf{X}]^{-1}$;

3. Calcule
$$\alpha(\boldsymbol{\beta}^{(t-1)}, \boldsymbol{\beta}^*) = \min\left\{1, \frac{p(\boldsymbol{\beta}^*|-)}{p(\boldsymbol{\beta}^{(t-1)}|-)}\right\},$$
   sendo $p(\boldsymbol{\beta}|-)$ a distribuição condicional completa do vetor $\boldsymbol{\beta}$;

4. Gere $u \sim U[0,1]$. Se $u < \alpha(\boldsymbol{\beta}^{(t-1)}, \boldsymbol{\beta}^*)$, faça $\boldsymbol{\beta}^{(t)} = \boldsymbol{\beta}^*$, caso contrário, faça $\boldsymbol{\beta}^{(t)} = \boldsymbol{\beta}^{(t-1)}$;

5. Tome $t = t + 1$ e volte ao passo 2 até que a amostra *a posteriori* desejada seja obtida após a convergência das cadeias.

No Apêndice B, o IWLS é detalhado para o modelo Poisson log-linear descrito no Capítulo 4.

O IWLS possui alta taxa de aceitação, pois o algoritmo WLS Bayesiano fornece boas aproximações a cada iteração, de modo que a medida que as iterações avançam, menos se rejeita. Um outro ponto interessante é que o IWLS, sob o paradigma Bayesiano, pode ser utilizado para estimar qualquer componente dos GLMM, seja ele fixo ou aleatório. Já sob a ótica da inferência Clássica, o WLS é utilizado apenas na estimação dos efeitos fixos, pois na estimação dos efeitos aleatórios é necessário o uso de métodos de aproximação como integração de Monte Carlo, métodos de quadratura entre outros.

Vale ressaltar que o IWLS tem restrições quanto a distribuição da quantidade de interesse. Isto é, a mesma precisa pertencer a família exponencial. Em casos em que a distribuição da quantidade de interesse é log-côncava mas não pertence a família exponencial, o Metropolis pode ser combinado com outros algoritmos de maximização. Em situações em que não é log-côncava, é possível usar o Metropolis ou outras versões do Metropolis como, por exemplo, o ARMS.

Na seção 2.2 de Chen et al. (2000), é dado um exemplo em que a distribuição da quantidade de interesse não é log-côncava, porém, é possível encontrar a distribuição de uma outra variável, que é função da variável original e que estabelece uma relação um-a-um, de modo que a distribuição da nova variável é log-côncava. Em outras palavras, ao ínves de amostrar diretamente da distribuição da quantidade de interesse original, pode-se amostrar da distribuição da nova variável. A partir disso, pode-se maximizar a distribuição da nova variável utilizando algum método de otimização como os algoritmos de Newton-Raphson ou de Nelder-Mead e, com isso, configurar a distribuição de propostas do Metropolis.

## 3.6 Metropolis Adaptativo

Haario et al. (2001) apresentaram o algoritmo Metropolis Adaptativo (AM), que utiliza toda informação da cadeia para configurar a distribuição de propostas, sendo ela Gaussiana. Devido a sua natureza adaptativa, o algoritmo não pode ser visto como um processo Markoviano, contudo, os autores provam que o mesmo possui as devidas propriedades de ergodicidade.

Em sua essência, o AM é baseado no RWM, o qual foi descrito anteriormente, e no algoritmo *Adaptive Proposal* (AP); ver Haario et al. (1999). Diferentemente do AP, onde a distribuição de propostas é centrada no estado atual da cadeia e a matriz de covariâncias é obtida a partir de um número fixo e finito de estados anteriores, o AM atualiza continuamente a matriz de covariâncias da distribuição de propostas utilizando todos os estados anteriores. De certa forma, o AM pode ser visto como uma versão do AP, salvo uma importante diferença: os autores não conseguiram provar as propriedades de ergodicidade do algoritmo AP.

Suponha que até a iteração $t-1$ do AM foram amostrados os estados $\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(t-1)}$, sendo $\boldsymbol{\theta}^{(0)} \in \mathbb{R}^d$ o estado inicial. A matriz de covariâncias da distribuição de propostas é

definida da seguinte maneira:

$$\boldsymbol{\Sigma}_t = \begin{cases} \boldsymbol{\Sigma}_0, & \text{se } t \leq t_0, \\ \boldsymbol{\Sigma}_{t-1} + s_d \times \varepsilon \times \mathbf{I}_d, & \text{se } t > t_0, \end{cases}$$

sendo $\boldsymbol{\Sigma}_0$ uma matriz positiva definida $d$-dimensional que deve expressar o conhecimento *a priori* que se tem sobre a estrutura de covariâncias da distribuição *a posteriori*, $\varepsilon > 0$ uma constante, $t_0 > 0$ o tamanho do período inicial, $\mathbf{I}_d$ uma matriz identidade de ordem $d$ e $s_d$ um parâmetro que depende somente da dimensão $d$ da matriz $\boldsymbol{\Sigma}_0$. Lembrando que a estimação da matriz de covariâncias tomando como base os vetores $\boldsymbol{\theta}^{(0)}$, $\boldsymbol{\theta}^{(1)}$, ..., $\boldsymbol{\theta}^{(t)} \in \mathbb{R}^d$ é dada da seguinte forma

$$\boldsymbol{\Sigma}_t = \text{cov}(\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(t)}) = \frac{1}{t}\left[\sum_{i=0}^{t} \boldsymbol{\theta}^{(i)}\boldsymbol{\theta}^{(i)^\top} - (t+1)\bar{\boldsymbol{\theta}}^{(t)}\bar{\boldsymbol{\theta}}^{(t)^\top}\right], \tag{3.7}$$

com $\bar{\boldsymbol{\theta}}^{(t)} = \frac{1}{(t+1)}\sum_{i=0}^{t} \boldsymbol{\theta}^{(i)}$ e $\boldsymbol{\theta}^{(i)} \in \mathbb{R}^d$ um vetor coluna. A matriz $\boldsymbol{\Sigma}_t$ é uma estimativa empírica da estrutura de covariâncias da distribuição alvo baseada em todos os valores amostrados até a $t$-ésima iteração. A partir disso, a fórmula de recursão da matriz de covariâncias satisfaz a equação

$$\boldsymbol{\Sigma}_{t+1} = \frac{t-1}{t}\boldsymbol{\Sigma}_t + \frac{s_d}{t}\left[t\bar{\boldsymbol{\theta}}^{(t-1)}\bar{\boldsymbol{\theta}}^{(t-1)^\top} - (t+1)\bar{\boldsymbol{\theta}}^{(t)}\bar{\boldsymbol{\theta}}^{(t)^\top} + \bar{\boldsymbol{\theta}}^{(t)}\bar{\boldsymbol{\theta}}^{(t)^\top} + \varepsilon\mathbf{I}_d\right]. \tag{3.8}$$

É importante destacar que a escolha do valor de $t_0$ impacta diretamente no efeito adaptativo do algoritmo. Isto é, quanto maior for $t_0$ mais lentamente a matriz de covariâncias empírica converge para a matriz de covariâncias da distribuição *a posteriori*. A constante $s_d = (2.4)^2/d$ foi adotada pois, de acordo com resultados teóricos e númericos mostrados em Gelman et al. (1996), ela otimiza a autocorrelação da cadeia gerada pelo Metropolis quando a distribuição de propostas e a distribuição alvo são Gaussianas. A constante $\varepsilon > 0$ é introduzida como uma medida de segurança no intuito de evitar que $\boldsymbol{\Sigma}_t$ se torne singular. Com isso, o algoritmo AM pode ser descrito como segue.

**Algoritmo** Metropolis AM

1. Inicie $t = 1$ e defina valores iniciais $\boldsymbol{\theta}^{(0)}$ e $\boldsymbol{\Sigma}_0$;
2. Amostre $\boldsymbol{\theta}^*$ da distribuição $N_d(\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\Sigma}_{t-1})$;
3. Calcule $\alpha(\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{p(\boldsymbol{\theta}^*|-)}{p(\boldsymbol{\theta}^{(t-1)}|-)}\right\}$;
4. Gere $u \sim U[0,1]$. Se $u < \alpha(\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\theta}^*)$, faça $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^*$, caso contrário, faça $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$;
5. Atualize a matriz de covariâncias $\boldsymbol{\Sigma}_t$ utilizando (3.7) ou (3.8);
6. Tome $t = t + 1$ e volte ao passo 2 até que a convergência seja atingida.

Uma outra versão do AM foi proposta por Roberts e Rosenthal (2009) e também será considerada no presente trabalho. Agora, a cada iteração $t$ no passo 2 do Metropolis AM, $\boldsymbol{\theta}^*$ é amostrado de uma $N_d[\boldsymbol{\theta}^{(t-1)}, (0.1^2/d)\mathbf{I}_d]$, se $t \leq 2d$, e

$$(1-\rho)N_d[\boldsymbol{\theta}^{(t-1)}, (2.4^2/d)\boldsymbol{\Sigma}_{t-1}] + \rho N_d[\boldsymbol{\theta}^{(t-1)}, (0.1^2/d)\mathbf{I}_d], \quad \text{se } t > 2d,$$

com $\rho \in [0,1]$. O componente $N_d[\boldsymbol{\theta}^{(t-1)}, (0.1^2/d)\mathbf{I}_d]$ é introduzido como uma medida de segurança para evitar que o algoritmo fique preso em valores problemáticos que podem deixar a matriz de covariâncias singular. Por uma questão de distinção, denotaremos AM-HA como sendo o AM proposto por Haario et al. (2001) e AM-RR como a versão proposta por Roberts e Rosenthal (2009).

## 3.7 Metropolis Adaptativo Robusto

O algoritmo Metropolis Adaptativo Robusto (RAM) (Vihola, 2012) possui características bastante interessantes, pois além de atualizar continuamente a matriz de covariâncias da distribuição de propostas, ele se atém a manter a taxa de aceitação em um nível predeterminado. Para isso, algumas restrições são feitas sobre a distribuição de propostas, que precisa ser esfericamente simétrica. Lembrando que um vetor $\boldsymbol{v}$ tem

distribuição esfericamente simétrica sobre $\boldsymbol{\theta}$ quando para toda matriz ortogonal $\boldsymbol{\Gamma}$ sua distribuição se mantém via multiplicação. Isto é,

$$\boldsymbol{\Gamma} \times (\boldsymbol{\upsilon} - \boldsymbol{\theta}) \stackrel{d}{=} \boldsymbol{\upsilon} - \boldsymbol{\theta},$$

sendo que "$\stackrel{d}{=}$" representa a igualdade em distribuição. Em paralelo, o vetor $\boldsymbol{\upsilon}$ possui distribuição esférica centralmente simétrica quando $\boldsymbol{\upsilon} - \boldsymbol{\theta} \stackrel{d}{=} \boldsymbol{\theta} - \boldsymbol{\upsilon}$ (Fang et al., 1990; Serfling, 2006).

A convergência do algoritmo é provada por meio de teoremas e proposições e estão disponíveis em Vihola (2012). A prova é feita assumindo que a distribuição geradora de propostas é Normal ou t-Student. No entanto, o autor não mostra que para outras distribuições esfericamente simétricas o algoritmo é válido. Durante a prova, a distribuição de propostas é completamente especificada, isto é, ela é $N_d(\mathbf{0}, \mathbf{I}_d)$ ou t-Student com $d$ graus de liberdade e parâmetros de locação e escala iguais a $\mathbf{0}$ e $\mathbf{I}_d$, respectivamente. Em paralelo, através de resultados empíricos, o autor mostra que o desempenho do RAM, quando a distribuição conjunta *a posteriori* não possui segundo momento finito, é estável se comparado com outros algoritmos adaptativos. Em situações onde a distribuição conjunta *a posteriori* possui segundo momento finito, o algoritmo mostrou-se atrativo.

**Algoritmo** Metropolis RAM

1. Inicie $t = 1$ e defina valores iniciais $\boldsymbol{\theta}^{(0)}$ e $\boldsymbol{\Sigma}_0$;

2. Faça $\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(t-1)} + \boldsymbol{\Sigma}_{t-1}\boldsymbol{u}_t$, sendo $\boldsymbol{u}_t \sim q()$ um vetor coluna gerado de uma distribuição esfericamente simétrica (Normal ou t-Student).

3. Calcule $\alpha(\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{p(\boldsymbol{\theta}^*|-)}{p(\boldsymbol{\theta}^{(t-1)}|-)}\right\}$;

4. Gere $u \sim U[0,1]$. Se $u < \alpha(\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\theta}^*)$, faça $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^*$, caso contrário, faça $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$;

5. Atualize a matriz de covariâncias satisfazendo a equação

$$\boldsymbol{\Sigma}_t\boldsymbol{\Sigma}_t^\top = \boldsymbol{\Sigma}_{t-1}\left[\mathbf{I}_d + \eta_t(\tau_t - \tau_*)\frac{\boldsymbol{u}_t\boldsymbol{u}_t^\top}{||\boldsymbol{u}_t||^2}\right]\boldsymbol{\Sigma}_{t-1}^\top.$$

6. Tome $t = t+1$ e volte ao passo 2 até obter a amostra desejada após a convergência da cadeia.

No algoritmo acima, considere: $\boldsymbol{\Sigma}_0$ a matriz de covariâncias *a priori*, $\boldsymbol{u}_t$ um vetor coluna $d$-dimensional, $\{\eta_t\}_{t\geq 1} \subset (0,1]$ uma sequência com termos que começam em 1 e decaem para zero, $\tau_t$ a taxa de aceitação na $t$-ésima iteração e $\tau_*$ a taxa de aceitação desejada. No caso da quantidade de interesse ser univariada, a matriz de covariâncias da distribuição de propostas, $\boldsymbol{\Sigma}_t = S_t$, é atualizada por

$$\log(S_t) = \log(S_{t-1}) + \frac{1}{2}\log(1 + \eta_t(\tau_t - \tau_*)).$$

## 3.8 Metropolis Adaptativo Variacional Bayesiano

Mbalawata et al. (2015) propuseram um novo algoritmo Metropolis adaptativo que tem distribuição de propostas Gaussiana e sua matriz de covariâncias é atualizada ao longo das iterações através de um procedimento chamado Filtro de Kalman Adaptativo Variacional Bayesiano (Särkkä e Nummenmaa, 2009; Särkkä e Hartikainen, 2013). De forma geral, o algoritmo Metropolis Adaptativo Variacional Bayesiano (*Variational*

*Bayesian Adaptive Metropolis* - VBAM) tem como objetivo fornecer amostras da distribuição *a posteriori*, a partir de suas distribuições condicionais completas, configurando a convergência da cadeia através da atualização da matriz de covariâncias da distribuição de propostas. Para que fique mais claro como a estimação da matriz de covariâncias da distribuição de propostas se dá, faremos uma breve descrição do Filtro de Kalman e do Filtro de Kalman Adaptativo Variacional Bayesiano (*Variational Bayesian Adaptive Kalman Filter* - VB-AKF). Posteriormente, descreveremos o algoritmo VBAM.

O Filtro Kalman é um algoritmo utilizado na previsão de estados futuros em modelos dinâmicos lineares Gaussianos. Em termos gerais, este modelo pode ser representado da seguinte maneira

$$\ddot{\mathbf{x}}_t \sim N(\mathbf{A}_{t-1}\ddot{\mathbf{x}}_{t-1}, \mathbf{Q}_{t-1}),$$
$$\ddot{\mathbf{y}}_t \sim N(\mathbf{H}_{t-1}\ddot{\mathbf{x}}_t, \mathbf{\Sigma}_t),$$

sendo $\mathbf{A}_{t-1}$ a matriz dinâmica do modelo, $\mathbf{Q}_{t-1}$ a matriz de covariâncias do ruído do processo, $\mathbf{H}_t$ a matriz de medição, $\mathbf{\Sigma}_t$ a matriz de covariâncias dos ruídos de medição, $\ddot{\mathbf{y}}_t \in \mathbb{R}^d$ o vetor de estados observados, $\ddot{\mathbf{x}}_t \in \mathbb{R}^n$ o vetor de estados desconhecidos (ou o vetor de estados dinâmicos), o qual tem-se interesse em fazer a estimação (ou predição). As matrizes $\mathbf{A}_{t-1}$, $\mathbf{Q}_{t-1}$, $\mathbf{H}_t$ e $\mathbf{\Sigma}_t$ são conhecidas e, assumindo $\ddot{\mathbf{x}}_0 \sim N(\mathbf{m}_0, \mathbf{P}_0)$, a estimação dos estados dinâmicos pode ser descrita, via Filtro de Kalman, como segue

1. Predição:

$$\mathbf{m}_t^- = \mathbf{A}_{t-1}\mathbf{m}_{t-1},$$
$$\mathbf{P}_t^- = \mathbf{A}_{t-1}\mathbf{P}_{t-1}\mathbf{A}_{t-1}^\top + \mathbf{Q}_{t-1}.$$

2. Atualização:

$$\mathbf{S}_t = \mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^\top + \mathbf{\Sigma}_t,$$
$$\mathbf{K}_t = \mathbf{P}_t^-\mathbf{H}_t^\top\mathbf{S}_t^{-1},$$
$$\mathbf{m}_t = \mathbf{m}_t^- + \mathbf{K}_t(\ddot{\mathbf{y}}_t - \mathbf{H}_t\mathbf{m}_t^-),$$
$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^\top,$$

com $\mathbf{m}_t^-$, $\mathbf{m}_t$ e $\mathbf{P}_t^-$, $\mathbf{P}_t$ sendo as médias e covariâncias dos estados dinâmicos *a priori* e *a posteriori*, respectivamente. O algoritmo VB-AKF pode ser visto como uma versão do Filtro de Kalman usual quando a matriz de covariâncias dos ruídos de medição $\mathbf{\Sigma}_t$ é desconhecida. Isto é,

$$\begin{aligned}
\ddot{\mathbf{x}}_t &\sim N(\mathbf{A}_{t-1}\ddot{\mathbf{x}}_{t-1}, \mathbf{Q}_{t-1}),\\
\ddot{\mathbf{y}}_t &\sim N(\mathbf{H}_{t-1}\ddot{\mathbf{x}}_t, \mathbf{\Sigma}_t),\\
\mathbf{\Sigma}_t &\sim p(\mathbf{\Sigma}_t|\mathbf{\Sigma}_{t-1}).
\end{aligned}$$

Nesse sentido, Särkkä e Nummenmaa (2009) e Särkkä e Hartikainen (2013) propuseram uma versão do Filtro de Kalman que estima $\mathbf{\Sigma}_t$ juntamente com os estados dinâmicos. Sob a ótica Bayesiana, o Filtro de Kalman usual estima os estados dinâmicos através da distribuição condicional completa de $\ddot{\mathbf{x}}_t$ dado todos os $\ddot{\mathbf{y}}_t$. Já no caso de $\mathbf{\Sigma}_t$ desconhecido, a ideia utilizada por Särkkä e Nummenmaa (2009) foi de obter uma aproximação da distribuição conjunta de $\mathbf{\Sigma}_t$ e $\ddot{\mathbf{x}}_t$ dado $\ddot{\mathbf{y}}_t$,

$$\begin{aligned}
p(\ddot{\mathbf{x}}_t|\ddot{\mathbf{y}}_{1:t}) &= N(\ddot{\mathbf{x}}_t|\mathbf{m}_t, \mathbf{P}_t),\\
p(\ddot{\mathbf{x}}_t, \mathbf{\Sigma}_t|\ddot{\mathbf{y}}_{1:t}) &\approx N(\ddot{\mathbf{x}}_t|\mathbf{m}_t, \mathbf{P}_t)\ WI(\mathbf{\Sigma}_t|\mathbf{V}_t, v_t),
\end{aligned}$$

com $\mathbf{m}_t$ e $\mathbf{P}_t$ obtidos via Filtro de Kalman usual e $\mathbf{V}_t$ e $v_t$ parâmetros da distribuição Wishart Inversa. Com isso, Särkkä e Hartikainen (2013) propuseram o algoritmo VB-AKF, que é utilizado para atualizar a matriz de covariâncias da distribuição de propostas do VBAM.

**Algoritmo** VB-AKF

1. Inicie $v_0$, $\mathbf{m}_0$, $\mathbf{P}_0$ e $\mathbf{\Sigma}_t$.

2. Para $t = 1, 2, 3, ...$

   (a) Predição:

$$\begin{aligned}
\mathbf{m}_t^- &= \mathbf{A}_{t-1}\mathbf{m}_{t-1}, \\
\mathbf{P}_t^- &= \mathbf{A}_{t-1}\mathbf{P}_{t-1}\mathbf{A}_{t-1}^\top + \mathbf{Q}_{t-1}, \\
v_t^- &= \rho(v_{t-1} - d - 1) + d + 1), \\
\mathbf{\Sigma}_t^- &= \mathbf{B}\mathbf{\Sigma}_{t-1}\mathbf{B}^\top.
\end{aligned}$$

   (b) Atualização: Tome $v_t = v_t^- + 1$, $\mathbf{\Sigma}_t^{(1)} = \mathbf{\Sigma}_t^-$ e $j = 1$,

$$\begin{aligned}
\mathbf{S}_t^{(j+1)} &= \mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^\top + \mathbf{\Sigma}_t^{(j)}, \\
\mathbf{K}_t^{(j+1)} &= \mathbf{P}_t^-\mathbf{H}_t^\top(\mathbf{S}_t^{(j+1)})^{-1}, \\
\mathbf{m}_t^{(j+1)} &= \mathbf{m}_t^- + \mathbf{K}_t^{(j+1)}(\ddot{\mathbf{y}}_t - \mathbf{H}_t\mathbf{m}_t^-), \\
\mathbf{P}_t^{(j+1)} &= \mathbf{P}_t^- - \mathbf{K}_t^{(j+1)}\mathbf{S}_t^{(j+1)}(\mathbf{K}_t^{(j+1)})^\top, \\
\mathbf{\Sigma}_t^{(j+1)} &= (\frac{v_{t-1} - d - 1}{v_t - d - 1})\mathbf{\Sigma}_t^- + (\frac{1}{v_t - d - 1})\mathbf{H}_t\mathbf{P}_t^{(j+1)}\mathbf{H}_t^\top \\
&\quad + (\frac{1}{v_t - d - 1})(\ddot{\mathbf{y}}_t - \mathbf{H}_t\mathbf{m}_t^{(j+1)})(\ddot{\mathbf{y}}_t - \mathbf{H}_t\mathbf{m}_t^{(j+1)})^\top. \qquad (3.9)
\end{aligned}$$

   (c) Tome $j = j + 1$ e volte ao passo (b) até a convergência ser atingida (para $j = 1, 2, ..., J$).

   (d) Tome $\mathbf{\Sigma}_t = \mathbf{\Sigma}_t^{(J)}$, $\mathbf{m}_t = \mathbf{m}_t^{(J)}$ e $\mathbf{P}_t = \mathbf{P}_t^{(J)}$.

3. Tome $t = t + 1$ e volte ao passo 2 até que a convergência de $\mathbf{\Sigma}_t$ seja obtida.

O valor de $J$ varia em cada problema. Para o estudo de simulação que será apresentado no Capítulo 5, verificamos que $J = 7$ foi suficiente. Isto é, para cada iteração $t$, foram necessárias 7 iterações até que a convergência da matriz $\mathbf{\Sigma}_t$ fosse observada. De acordo com os autores, algumas restrições são feitas de modo a garantir a convergência do VB-

AKF tais como $\mathbf{B} = \mathbf{I}$, $\rho = 1$ e $a\mathbf{I} \leq \boldsymbol{\Sigma}_t \leq b\mathbf{I}$, sendo $a$ uma constante muito pequena e $b$ uma constante grande tal que $0 < a < b < \infty$; os autores não recomendam valores deixando a cargo do usuário a definição dos mesmos (neste trabalho adotamos $a = 0{,}00001$ e $b$ =10.000). Note que, nesse contexto, $\boldsymbol{\Sigma}_t \leq b\mathbf{I}$ e $a\mathbf{I} \leq \boldsymbol{\Sigma}_t$ significa que $b\mathbf{I} - \boldsymbol{\Sigma}_t \geq 0$ e $\boldsymbol{\Sigma}_t - a\mathbf{I} \geq 0$ são matrizes positivas semidefinidas. É importante ressaltar que o algoritmo VB-AKF é usado somente para atualizar a matriz de covariâncias da distribuição de propostas. O algoritmo VBAM é descrito como segue.

---

**Algoritmo** Metropolis VBAM

1. Inicie $t = 1$, $\boldsymbol{\Sigma}_0$, $\mathbf{P}_0$, $\lambda_0$ e $\boldsymbol{\theta}^{(0)}$;

2. Amostre $\boldsymbol{\theta}^*$ da distribuição $N(\boldsymbol{\theta}^{(t-1)}, \lambda_{t-1}\ \boldsymbol{\Sigma}_{t-1})$;

3. Calcule
$$\alpha(\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{p(\boldsymbol{\theta}^*|-)}{p(\boldsymbol{\theta}^{(t-1)}|-)}\right\}.$$

4. Gere $u \sim U[0,1]$. Se $u < \alpha(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^*)$, faça $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^*$, caso contrário, faça $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$;

5. Atualize a matriz de covariâncias usando o VB-AKF, com $\ddot{\mathbf{y}}_t = \boldsymbol{\theta}^{(t-1)}$. Verifique se $a\mathbf{I} < \boldsymbol{\Sigma}_t < b\mathbf{I}$. Caso contrário, tome $\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_{t-1}$ e faça uma única iteração do VB-AKF, no passo de "Atualização", ignorando a última equação (3.9) referente a $\boldsymbol{\Sigma}_t^{(j+1)}$;

6. Tome $t = t + 1$ e retorne ao passo 2 até que a amostra *a posteriori* desejada seja obtida após a convergência.

---

Opcionalmente, é possível atualizar $\lambda_{t-1}$, no entanto, neste trabalho fixamos $\lambda_{t-1} = (2,4)^2/d$ para todo $t$ (a convergência é provada para $\lambda_{t-1}$ fixo e não fixo). Assim como no algoritmo AM, a constante $\lambda_{t-1} = (2,4)^2/d$ foi adotada pois, sob certas condições, previamente mencionadas na Seção 3.6, ela otimiza a autocorrelação da cadeia gerada pelo Metropolis. Ressaltamos que as demonstrações sobre a convergência dos algoritmos

VB-AKF e VBAM podem ser encontradas em Mbalawata et al. (2015).

## 3.9 Método da rejeição adaptativo com Metropolis

Antes de introduzir o método da rejeição adaptativo com Metropolis (*Adaptive Rejection Metropolis Sampling* - ARMS), iremos descrever o método da rejeição (*Rejection Sampling* - RS) e o método da rejeição adaptativo (*Adaptive Rejection Sampling* - ARS), pois, de certa forma, o ARMS pode ser visto como uma extensão de ambos os métodos. Vale destacar que os métodos apresentados nesta seção não serão o foco da comparação mostrada no Capítulo 5. Na ocasião, o ARS será usado especificamente para gerar os efeitos aleatórios, substituindo o passo Metropolis que tem custo computacional maior, e o vetor de coeficientes $\boldsymbol{\beta}$ será gerado usando os métodos Metropolis-Hastings apresentados.

### 3.9.1 Método da rejeição

O RS é um método que permite obter amostras independentes de uma dada distribuição de interesse, $p(\theta_k|\boldsymbol{\theta}_{-k})$, mesmo em situações onde conhecemos apenas o núcleo da mesma. Inicialmente proposto por (Ripley, 1987), o RS gera amostras da distribuição de interesse através de amostragem indireta. Para gerar valores de $p(\theta_k|\boldsymbol{\theta}_{-k})$, o RS requer a especificação de uma distribuição $q(\theta_k)$, da qual seja fácil amostrar, tal que $p(\theta_k|\boldsymbol{\theta}_{-k}) \leq cq(\theta_k)\ \forall\ \theta_k \in \Theta$, sendo $c < \infty$ uma constante e $\Theta$ o domínio da funcão $p(\theta_k|\boldsymbol{\theta}_{-k})$. Em termos gerais, o RS tem os seguintes passos:

**Algoritmo** RS

1. Amostre $\theta_k^*$ de $q(\theta_k)$, com $k = 1, ..., d$;
2. Amostre $u \sim U[0, 1]$;
3. Se $u \leq \frac{p(\theta_k^*|\boldsymbol{\theta}_{-k})}{cq(\theta_k^*)}$, então $\theta_k = \theta_k^*$. Caso contrário, volte ao passo 1 até que um $\theta_k^*$ seja aceito.
4. Tome $k = k+1$ e repita os passos 1-3 até obter uma de amostra de tamanho $t$ para cada $\theta_k$.

A distribuição $q()$ é geralmente chamada de envelope e $c$ de constante do envelope. Sua escolha é de fundamental importância para eficiência do algoritmo (Gamerman e Lopes, 2006). É desejável que $q()$ seja uma distribuição similar a $p(\theta_k|\boldsymbol{\theta}_{-k})$, pois caso seja muito diferente, a constante $c$ deverá ser maior do que 1 para que a relação $p(\theta_k|\boldsymbol{\theta}_{-k}) \leq cq(\theta_k)$ seja garantida. O valor de $c$ está diretamente ligado a probabilidade de aceitação do RS; quanto maior $c$, menor é essa probabilidade. Por outro lado, valores pequenos levam a uma maior taxa de aceitação (Gamerman e Lopes, 2006).

### 3.9.2 Método da rejeição adaptativo

Sob certas restrições, o algoritmo ARS (Gilks e Wild, 1992; Gilks, 1992) pode ser visto como uma versão mais eficiente do RS. A eficiência (e a restrição) se dá pelo fato do ARS assumir que o logaritmo da função de interesse, $h(\theta_k) = log(p(\theta_k|\boldsymbol{\theta}_{-k}))$, é côncava, sendo $h(\theta_k)$ univariada. O fato da função de interesse ser log-côncava se faz necessário, pois através da derivada da mesma é criada uma função limiar que faz com que a probabilidade de aceitação do algoritmo aumente a medida que o número de iterações cresce.

Seja $\mathcal{P}$ o conjunto de pontos ordenados $\tilde{\theta}_p$, com $p = 0, 1, ..., r+1$, que pertencem ao domínio da função $p(\theta_k|\boldsymbol{\theta}_{-k})$. Considere $L_{p,p+1}(\theta_k)$ a reta entre os pontos $(\tilde{\theta}_p, h(\tilde{\theta}_p))$ e $(\tilde{\theta}_{p+1}, h(\tilde{\theta}_{p+1}))$. Para $\theta_k \in [\tilde{\theta}_p, \tilde{\theta}_{p+1}]$, assuma

$$m_r(\theta_k) = \min\{L_{p-1,p}(\theta_k), L_{p+1,p+2}(\theta_k)\}. \tag{3.10}$$

Caso $\theta_k \in [\tilde{\theta}_p, \tilde{\theta}_{p+1}]^c$,

$$m_r(\theta_k) = \min\{L_{0,1}(\theta_k), L_{r,r+1}(\theta_k)\}.$$

Uma vez que $p(\theta_k|\boldsymbol{\theta}_{-k})$ é log-côncava em todo o seu domínio, temos que $m_r(\theta_k)$ é uma funcão que funciona como "envelope" para $h()$, isto é, $h(\theta_k) \leq m_r(\theta_k)$. Dessa forma, o ARS pode ser descrito da seguinte forma:

---

**Algoritmo** ARS

1. Inicie $\mathcal{P}$;
2. Gere $\theta_k^* \sim q(\theta_k) = \frac{e^{m_r(\theta_k)}}{\int_\Theta e^{m_r(\theta_k)}\, d\theta_k}$ e $u \sim U[0,1]$;
3. Se $u \leq \frac{p(\theta_k^*|\boldsymbol{\theta}_{-k})}{e^{m_r(\theta_k^*)}}$, então $\theta_k = \theta_k^*$. Caso contrário, faça $\mathcal{P} = \mathcal{P} \cup \{\theta_k^*\}$, reordene $\mathcal{P}$ e volte ao passo 2 até que $\theta_k^*$ seja aceito.
4. Repita os passos 1-3 até obter uma de amostra de tamanho $t$ para cada $\theta_k$.

---

Note que $q(\theta_k)$ é uma distribuição exponencial por partes (Robert e Casella, 2004), a partir da qual podemos amostrar da seguinte forma:

1. Selecione o intervalo $[\tilde{\theta}_p, \tilde{\theta}_{p+1}]$ com probabilidade

$$\frac{e^{\alpha_p \tilde{\theta}_{p+1} + b_p} - e^{\alpha_p \tilde{\theta}_p + b_p}}{\alpha_p \digamma},$$

sendo que $b_p$ e $\alpha_p$ são, respectivamente, o intercepto e o coeficiente angular da reta entre os pontos $(\tilde{\theta}_p, h(\tilde{\theta}_p))$ e $(\tilde{\theta}_{p+1}, h(\tilde{\theta}_{p+1}))$. Temos também que $\alpha_-$ e $b_-$ são a inclinação e o intercepto da reta formada pelos pontos $(\tilde{\theta}_-, h(\tilde{\theta}_-))$ e $(\tilde{\theta}_0, h(\tilde{\theta}_0))$, com $\theta_-$ sendo o menor valor no domínio de $p(\theta_k|\boldsymbol{\theta}_{-k})$. Já $\alpha_*$ e $b_*$ são inclinação e o intercepto da reta formada pelos pontos $(\tilde{\theta}_{r+1}, h(\tilde{\theta}_{r+1}))$ e $(\tilde{\theta}_*, h(\tilde{\theta}_*)$, com $\theta_*$ sendo o maior valor no domínio de $p(\theta_k|\boldsymbol{\theta}_{-k})$. Consequentemente,

$$\begin{aligned}
\digamma &= \int_{-\infty}^{\tilde{\theta}_0} e^{\alpha_- \theta_k + b_-}\, d\theta_k + \sum_{p=0}^{r} \int_{\tilde{\theta}_p}^{\tilde{\theta}_{p+1}} e^{\alpha_p \theta_k + b_p}\, d\theta_k + \int_{\tilde{\theta}_{p+1}}^{+\infty} e^{\alpha_* \theta_k + b_*}\, d\theta_k \\
&= \frac{e^{\alpha_- \tilde{\theta}_0 + b_-}}{\alpha_-} + \sum_{p=0}^{r} \frac{e^{\alpha_p \tilde{\theta}_{p+1} + b_p} - e^{\alpha_p \tilde{\theta}_p + b_p}}{\alpha_p} - \frac{e^{\alpha_* \tilde{\theta}_{r+1} + b_*}}{\alpha_{r+1}}.
\end{aligned}$$

2. Gere $u \sim U[0,1]$ e calcule

$$\theta_k^* = \frac{\log[e^{\alpha_p \tilde{\theta}_p} + u(e^{\alpha_p \tilde{\theta}_{p+1}} - e^{\alpha_p \tilde{\theta}_p})]}{\alpha_p}.$$

Vale destacar que a primeira versão do ARS, proposta por Gilks e Wild (1992), é um pouco diferente do algoritmo apresentado acima. Nela, o ARS possui duas funções que limitam $h(\theta_k)$ inferior e superiormente com retas tangentes e secantes sendo utilizadas para aproximar a função $h$.

Além disso, o conjunto inicial $\mathcal{P}$ pode ser formado por 2 a 4 pontos. Sempre que o valor $\theta_k^*$ for rejeitado, a função $m_r(\theta_k)$ é atualizada de modo a se aproximar mais da função $h(\theta_k)$, reduzindo assim o número de iterações $t$. Para que o ARS possa ser usado é essencial que a função $h(\theta_k)$ seja log-côncava (e univariada). Caso não seja, a função que limita $h(\theta_k)$ superiormente não será "envelope", ponto necessário no RS (Gamerman e Lopes, 2006). Na Tabela 2 de Gilks (1992), são listadas algumas distribuições de probabilidade que são log-côncavas com relação aos seus parâmetros.

### 3.9.3 Método da rejeição adaptativo com Metropolis

O ARMS foi proposto por (Gilks et al., 1995) como uma generalização do ARS para amostrar valores de distribuições univariadas que não são log-côncavas. Assim como o ARS, o ARMS atualiza a distribuição de propostas, $q()$, a medida que o número de iterações aumenta. Uma vez que o ARMS não se restringe ao fato da função de interesse $p(\theta_k|\boldsymbol{\theta}_{-k})$ ser log-côncava, ele configura a distribuição de propostas através de uma função que não necessariamente cobre toda a $p(\theta_k|\boldsymbol{\theta}_{-k})$ como no ARS. Isto é, agora $h(\theta_k) \leq m_r'(\theta_k)$ pode não ser verdade $\forall\ \theta_k \in \Theta$. Com isso, considere a seguinte função

$$m_r'(\theta_k) = \max\{L_{p,p+1}(\theta_k), [\min\{L_{p-1,p}(\theta_k), L_{p+1,p+2}(\theta_k)\}]\},\ \text{se } \tilde{\theta}_p \leq \theta_k < \tilde{\theta}_{p+1}. \quad (3.11)$$

Nos casos em que $\theta_k < \tilde{\theta}_0$, $m_r'(x) = L_{0,1}(x)$. Para $\theta_k \geq \tilde{\theta}_{r+1}$, temos $m_r'(\theta_k) = L_{r,r+1}(\theta_k)$. Com isso, o ARMS pode ser descrito como segue:

**Algoritmo** ARMS

1. Faça t = 1 e inicie $\mathcal{P}$ de modo independente de $\theta_k^{(t-1)}$, com $k = 1, ..., d$;

2. Gere $\theta_k^* \sim q(\theta_k) = \frac{e^{m'_r(\theta_k)}}{\int_\Theta e^{m'_r(\theta_k)}}$ e $u \sim U[0,1]$;

3. (a) Se $u > \frac{p(\theta_k^*|\boldsymbol{\theta}_{-k})}{e^{m'_r(\theta_k^*)}}$, faça $\mathcal{P} = \mathcal{P} \cup \{\theta_k^*\}$, reordene $\mathcal{P}$ e volte ao passo 2 até que $\theta_k^*$ seja aceito.

   (b) Se $u \leq \frac{p(\theta_k^*|\boldsymbol{\theta}_{-k})}{e^{m'_r(\theta_k^*)}}$, gere novamente $u \sim U[0,1]$;

   Se $u \leq \min\left\{1, \frac{p(\theta_k^*|\boldsymbol{\theta}_{-k}) \min[p(\theta_k^{(t-1)}|\boldsymbol{\theta}_{-k}), \exp\{m'_r(\theta_k^{(t-1)})\}]}{p(\theta_k^{(t-1)}|\boldsymbol{\theta}_{-k}) \min[p(\theta_k^*|\boldsymbol{\theta}_{-k}), \exp\{m'_r(\theta_k^*)\}]}\right\}$, então $\theta_k^{(t)} = \theta_k^*$. Caso contrário, $\theta_k^{(t)} = \theta_k^{(t-1)}$.

4. Repita os passos 1-3 até obter uma amostra de tamanho $t$ para cada $\theta_k$.

Diferentemente do ARS, o ARMS não gera amostras independentes. Isso se dá pela introdução do passo Metropolis que garante que os valores amostrados sejam provenientes de $p(\theta_k|\boldsymbol{\theta}_{-k})$, mesmo quando $h(\theta_k) > m'_r(\theta_k)$. Em situações onde a distribuição de interesse é log-côncava, a equação (3.11) se reduz a (3.10) e o ARS é retomado. Além disso, note que o ARMS possui dois testes de aceitação. O primeiro é feito no passo 3(a), e consiste em verificar se o valor que está sendo proposto seria pré-aceito como valor candidato. Sendo pré-aceito, o valor candidato passa por um outro teste em 3(b). Caso seja rejeitado em 3(a), o mesmo é usado para configurar a distribuição de propostas, através de sua inclusão ao conjunto $\mathcal{P}$. Agora, caso seja rejeitado em 3(b), o valor não é incluído ao conjunto $\mathcal{P}$. A princípio, pode-se pensar em incluir os pontos rejeitados em 3(b) ao conjunto $\mathcal{P}$ como uma alternativa para configurar a distribuição de propostas mais rapidamente, porém, dessa forma, o número de pontos inseridos em $\mathcal{P}$ pode ser alto, encarecendo o algoritmo. Com esta estratégia, do ponto de vista teórico, não é possível garantir a convergência do algoritmo; veja Martino et al. (2015).

Em Martino et al. (2015), são apresentados dois algoritmos que, de acordo com os autores, podem ser vistos como versões melhoradas do ARMS. O primeiro, chamado

de $A^2$RMS, melhora o desempenho do ARMS dando a possibilidade dos valores aceitos no passo 3(b) serem incluídos no conjunto $\mathcal{P}$. A ideia é uma tentativa de configurar a distribuição de propostas mais rapidamente do que no ARMS. O segundo algoritmo, chamado de I$A^2$RMS, configura a distribuição de proposta através de uma estrutura adaptativa, que leva em consideração os estados anteriores gerados, exceto o último; para mais detalhes veja Martino et al. (2015).

## 3.10 Critério de comparação

Os métodos MCMC descritos nas seções anteriores geram cadeias onde os estados possuem uma relação de dependência Markoviana; isto é, eles não são amostras i.i.d. da distribuição alvo. Essa estrutura de dependência tem efeito significativo sobre as estimativas intervalares, e precisa ser devidamente levada em consideração. Nesse sentido, suponha uma sequência de variáveis aleatórias $z_1, z_2, ..., z_t$ que podem ou não ser i.i.d.. O coeficiente de autocorrelação de *lag* $l$ é dado por

$$\hat{\rho}_l = \frac{\sum_{i=l+1}^{t}(z_i - \bar{z})(z_{i-l} - \bar{z})}{\sum_{i=1}^{t}(z_i - \bar{z})^2}, \text{ com } \bar{z} = \frac{\sum_{i=1}^{t} z_i}{t}. \tag{3.12}$$

Em uma amostra i.i.d., a expressão em (3.12) é (aproximadamente) zero. Além disso, considere a seguinte estatística para avaliar o tamanho efetivo da amostra de uma cadeia:

$$n_{ef} = \frac{t}{(1 + 2\sum_{l=1}^{L}\hat{\rho}_l)}, \tag{3.13}$$

com $t$ sendo o tamanho da cadeia e $\hat{\rho}_l$ o coeficiente de autocorrelação para $l = 1, 2, ..., L$, e $L$ suficientemente grande (nesse trabalho adotamos $L = 50$). Na prática, quanto maior for o denominador da fórmula em (3.13), mais autocorrelacionados são os estados da cadeia e menos eficiente é o método que gera valores candidatos. Por exemplo, suponha que o denominador é igual a 4, isto é, temos $n_{ef} = t/4$. Nesse caso, a cadeia deve ter $4t$ observações no intuito de representar (em termos de informação) uma amostra i.i.d. de tamanho $t$; para mais detalhes veja Gamerman e Lopes (2006) e Robert e Casella (2004). No Capítulo 5, a estatística $n_{ef}$ será utilizada como critério de comparação entre os algoritmos envolvendo o Metropolis-Hastings que estamos estudando.

## 3.11 Conclusões do capítulo

Nesse capítulo apresentamos os métodos MCMC que serão comparados em um cenário de simulação mais adiante, no Capítulo 5. De forma sucinta, os métodos Metropolis-Hastings introduzidos aqui fornecem amostras da distribuição de interesse, que são obtidas de forma indireta, e se diferenciam com relação a especificação da matriz de covariâncias da distribuição de propostas.

Por exemplo, no RWM essa matriz (ou o escalar) de covariâncias da distribuição de propostas é configurada inicialmente e não muda durante as iterações do algoritmo. Em muitos casos, pode-se observar uma baixa taxa de aceitação e/ou convergência lenta para a distribuição de interesse.

Já o Metropolis IWLS configura tanto a média quanto a matriz de covariâncias da distribuição de propostas. Ele pode ser visto como uma generalização do WLS Bayesiano, usado para fazer estimação dos coeficientes de regressão em GLM. Lembrando que o IWLS pode ser utilizado somente quando a quantidade de interesse tem distribuição na família exponencial.

Diferentemente do RWM, o AM-HA e o AM-RR configuram a matriz de covariâncias calculando a covariância dos estados da cadeia gerada; isto é, são algoritmos adaptativos ao longo das iterações. Assim como o AM-HA e o AM-RR, o RAM também configura a matriz de covariâncias de forma adaptativa, com uma única diferença: é possível especificar sua taxa de aceitação.

Por outro lado, o VBAM configura a matriz de covariâncias da distribuição de propostas através do algoritmo VB-AKF, que pode ser visto como versão do filtro de Kalman quando a matriz de covariâncias dos ruídos de medição é desconhecida. Em essência, o VBAM é um algoritmo que requer um outro algoritmo, esse último específico para atualizar a matriz de covariâncias da distribuição de propostas.

Além disso, embora também seja um algoritmo adaptativo, o ARMS não requer uma parte específica para atualizar a distribuição de propostas uma vez que a medida que o número de iterações aumenta, menos valores candidatos são rejeitados devido ao uso de segmentos de retas para aproximar a distribuição de interesse. Vale lembrar que

diferentemente do ARS, o ARMS não se restringe a amostrar de distribuições que sejam log-côncavas.

# Capítulo 4

# Modelo Poisson log-linear

Este capítulo apresentará a descrição dos dados que serão considerados na aplicação real. Posteriormente, serão introduzidos o modelo Poisson log-linear no contexto de dados longitudinais de contagem, bem como a função de verossimilhança, as distribuições *a priori* e as distribuições condicionais completas.

## 4.1 Descrição dos dados

A epilepsia é uma desordem neurológica caracterizada por recorrentes e repentinos ataques, espasmos e convulsões com ou sem perda de consciência. Os dados utilizados neste trabalho são de um estudo clínico realizado com 59 pessoas portadoras desta doença. Esses pacientes foram separados aleatoriamente em dois grupos para receber placebo ou uma nova droga chamada de *Progabide*, que tem como objetivo reduzir o número de ataques epilépticos. O número de ataques ocorridos para cada paciente nas duas semanas anteriores é registrado em cada uma das 4 visitas à clínica; o tempo entre cada uma das visitas é de duas semanas. Além disso, o estudo também registrou informações sobre a idade de cada paciente e o número de ataques durante um período de 8 semanas que antecedeu o início do estudo. Essa última variável é chamada de *baseline*.

Esses dados já foram analisados na literatura. Thall e Vail (1990) introduziram alguns modelos de covariância para lidar com problemas de dados longitudinais com sobredispersão. Adicionalmente, Fotouhi (2008) sugeriu diferentes modelos que incluem e

excluem, basicamente, dois componentes: efeitos aleatórios e correlação serial. A inclusão ou exclusão desses componentes é feita para lidar com o problema de sobredispersão verificado no estudo clínico descrito anteriormente. Em resumo, as análises conduzidas por Fotouhi (2008) sugerem que o modelo que inclui efeitos aleatórios é o mais apropriado para modelar os dados.

Cinco variáveis serão incluídas na análise que será feita no Capítulo 6. No entanto, as mesmas também serão utilizadas no cenário de simulação.

- $A_i$ representa o logaritmo da idade (em anos) do $i$-ésimo paciente;
- $B_i$ é o logaritmo da média da variável *baseline* dado por $\log[\frac{1}{4}$ (*baseline*)]. O valor do período *baseline* é dividido por 4 para que se tenha uma média do número de convulsões a cada duas semanas nesse período (de 8 semanas) que antecedeu o início do estudo. Dessa forma, pode-se compará-lo com o número de ataques registrados em cada uma das 4 visitas.
- $T_i$ indica o tratamento aplicado ao $i$-ésimo paciente (1 = *Progabide*, 0 = placebo);
- $V4_{iv}$ é o indicador do $i$-ésimo paciente para quarta visita a clínica (1 se $v = 4$, 0 caso contrário);
- $TB_i$ denota a interação entre as variáveis tratamento e o logaritmo da média da variável *baseline* ($T_i$ e $B_i$).

As Tabelas 4.1, 4.2 e 4.3 apresentam algumas estatísticas descritivas que sugerem possível sobredispersão no estudo clínico com pacientes epilépticos. A Tabela 4.1 indica sobredispersão nas visitas, nos grupos placebo e *Progabide* e nos dados completos. Note que as razões entre as variâncias e as médias são maiores do que 1. A Tabela 4.2 mostra a existência de alta variabilidade nos grupos placebo e *Progabide* ao longo das visitas. É possível observar que a razão variância/média é maior entre os pacientes que utilizaram *Progabide*. A Tabela 4.3 indica sobredispersão no período *baseline* por grupo de tratamento; embora as médias sejam similares, as razões entre variâncias e médias são bem maiores do que 1.

Com isso, pelo fato de $Y_{iv}$ (número de ataques epilépticos) se tratar de uma contagem, intuitivamente, pode-se pensar no modelo de Poisson sem efeitos aleatórios, no entanto, este modelo assume que a média é igual a variância. Se simplesmente desconsiderarmos a sobredispersão, as estimativas dos coeficientes da regressão podem ser afetadas. Em situações como essa, a variância dos coeficientes da regressão é subestimada, os resíduos do modelo são grandes e, em alguns casos, isso leva à conclusão de coeficientes não significativos (Fotouhi, 2008). Em outras palavras, se a sobredispersão não for devidamente tratada, decisões equivocadas podem ser tomadas.

| Estatística | Visita 1 | Visita 2 | Visita 3 | Visita 4 | Placebo | Progabide | Dados completos |
|---|---|---|---|---|---|---|---|
| Média | 8.95 | 8.36 | 8.44 | 7.31 | 8.61 | 7.97 | 8.27 |
| Var./Média | 24.59 | 12.42 | 23.72 | 12.75 | 12.54 | 24.34 | 18.45 |

Tabela 4.1: Estatísticas de ataques epilépticos por visitas e tratamentos (Fotouhi, 2008).

| | Placebo | | Progabide | |
|---|---|---|---|---|
| Visita | Média | Var./Média | Média | Var./Média |
| 1 | 9.36 | 10.98 | 8.58 | 38.78 |
| 2 | 8.29 | 8.04 | 8.42 | 16.71 |
| 3 | 8.79 | 24.5 | 8.13 | 23.75 |
| 4 | 7.96 | 7.31 | 6.71 | 18.92 |

Tabela 4.2: Estatísticas de ataques epilépticos por visita *versus* tratamento (Fotouhi, 2008).

| Estatística | Baseline | Placebo | Progabide |
|---|---|---|---|
| Média | 31.24 | 30.79 | 31.65 |
| Var./Média | 22.14 | 22.13 | 24.79 |

Tabela 4.3: Estatísticas de tratamentos (placebo e *Progabide*) para o período de oito semanas que antecedeu o início do estudo (Fotouhi, 2008).

## 4.2 Modelo Poisson log-linear

Um modelo hierárquico será explorado neste trabalho. Suponha que a variável res-

posta $Y_{iv} \sim \text{Poisson}(\mu_{iv})$ representa o número de casos relacionados a um determinado indivíduo $i = \{1, 2, ..., N\}$ no tempo $v = \{1, 2, ..., M\}$. Note que $Y_{iv}$ assume apenas valores inteiros não negativos. A distribuição de Poisson é membro da família exponencial; ou seja, ela pode ser escrita da forma em (3.4). Neste caso, a função de ligação canônica é $g(\mu_{iv}) = \eta_{iv} = \log(\mu_{iv})$. Em seguida, defina o vetor $\mathbf{y}_i = (Y_{i1}, Y_{i2}, ..., Y_{iM})^\top$ contendo todas as respostas relacionadas ao indivíduo $i$ e $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N)^\top$ representando todas as respostas de todos os indivíduos.

Agora, denote por $\mathbf{X}$ a matriz de dados contendo todas as covariáveis que estão a disposição para análise. O vetor $\mathbf{x}_{iv} = (1, x_{iv2}, x_{iv3}, ..., x_{ivd})$ representa a linha da matriz $\mathbf{X}$ associada ao indivíduo $i$ no tempo $v$. Note que $d$ variáveis explicativas são definidas no modelo e o valor 1 é inserido na primeira coluna para especificar o intercepto. Por fim, o modelo que associa $\mu_{iv}$ às covariáveis é dado por:

$$log(\mu_{iv}) = \mathbf{x}_{iv}\boldsymbol{\beta} + \gamma_i + \delta_{iv}, \tag{4.1}$$

sendo $\boldsymbol{\beta} = (\beta_0, \beta_2, ..., \beta_d)^\top$ e assumindo $\gamma_i \sim N(0, \sigma^2_\gamma)$ e $\delta_{iv} \sim N(0, \sigma^2_\delta)$. É importante ressaltar que tanto $\gamma_i$ quanto $\delta_{iv}$ são independentes uns dos outros e $\forall\ i \neq j$ e $v = (1, 2, ..., M)$, $\gamma_i$ é independente de $\gamma_j$ e $\delta_{iv}$ é independente de $\delta_{jv}$.

Os dois componentes $\gamma_i$ e $\delta_{iv}$ são adicionados ao preditor linear para lidar com a sobredispersão observada nos dados descritos na Seção 4.1. Ambos os componentes são efeitos aleatórios. De forma bem simples, o efeito aleatório $\gamma_i$ é introduzido a nível de indivíduo no intuito de capturar o efeito de características que não foram observadas, mas são observavéis e/ou não-mensuráveis. Isto é, diferentes indivíduos podem ser afetados, de alguma forma, por outros fatores que não foram considerados na análise. A mesma interpretação pode ser estendida ao componente $\delta_{iv}$, só que agora as características que afetam o indivíduo $i$ estão mudando ao longo do tempo. É importante ressaltar que o presente trabalho não tem como objetivo propor um novo modelo para analisar os dados acima descritos; o modelo Poisson log-linear já foi utilizado por outros autores na análise do mesmo problema (Thall e Vail, 1990; Fotouhi, 2008).

As seguintes distribuições *a priori* são assumidas: $\boldsymbol{\beta} \sim N_6(\mathbf{0}, \mathbf{V})$, $\sigma^2_\gamma \sim GI(g_1, g_2)$, $\sigma^2_\delta \sim GI(d_1, d_2)$, sendo que GI representa a distribuição Gama Inversa. Denote $\boldsymbol{\gamma} =$

$(\gamma_1, \gamma_2, ..., \gamma_N)^\top$, $\boldsymbol{\delta} = (\delta_{11}, \delta_{12}, ..., \delta_{NM})^\top$, $\boldsymbol{\gamma}_{-i}$ e $\boldsymbol{\delta}_{-(iv)}$, onde $\boldsymbol{\gamma}_{-i}$ e $\boldsymbol{\delta}_{-(iv)}$ representam os vetores $\boldsymbol{\gamma}$ e $\boldsymbol{\delta}$ sem o componente $\gamma_i$ e $\delta_{iv}$, respectivamente.

## 4.3 Distribuições condicionais completas

A distribuição *a posteriori* conjunta de $(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \sigma^2_\gamma, \sigma^2_\delta)$ não pode ser tratada analiticamente. Uma possível alternativa é usar a estrutura do algoritmo *Gibbs Sampling* para amostrar da distribuição alvo do parâmetro de interesse a partir das distribuições condicionais completas. Considere a função de verossimilhança do modelo Poisson log-linear:

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}) &= \prod_{i=1}^{N}\prod_{v=1}^{M} \frac{\mu_{iv}^{Y_{iv}} \exp\{-\mu_{iv}\}}{Y_{iv}!} \\ &= \prod_{i=1}^{N}\prod_{v=1}^{M} \frac{1}{Y_{iv}!} \exp\{Y_{iv}(\mathbf{x}_{iv}\boldsymbol{\beta} + \gamma_i + \delta_{iv})\} \times \\ &\times \exp\{-e^{\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv}}\} \\ &= \exp\{\sum_{i=1}^{N}\sum_{v=1}^{M} Y_{iv}(\mathbf{x}_{iv}\boldsymbol{\beta} + \gamma_i + \delta_{iv})\} \times \\ &\times \exp\{-\sum_{i=1}^{N}\sum_{v=1}^{M} e^{\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv}}\} \left(\prod_{i=1}^{N}\prod_{v=1}^{M} \frac{1}{Y_{iv}!}\right). \end{aligned} \tag{4.2}$$

As distribuições condicionais completas para $\boldsymbol{\beta}$,$\gamma_i$ e $\delta_{iv}$ são dadas por:

$$\begin{aligned} p(\boldsymbol{\beta}|\boldsymbol{\gamma}, \boldsymbol{\delta}, \sigma^2_\gamma, \sigma^2_\delta, \mathbf{y}, \mathbf{X}) &\propto p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}) \times p(\boldsymbol{\beta}) \\ &\propto \exp\{\sum_{i=1}^{N}\sum_{v=1}^{M} Y_{iv}(\mathbf{x}_{iv}\boldsymbol{\beta} + \gamma_i + \delta_{iv})\} \times \\ &\times \exp\{-\sum_{i=1}^{N}\sum_{v=1}^{M} e^{\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv}} - \frac{1}{2}\boldsymbol{\beta}^\top \mathbf{C}^{-1}\boldsymbol{\beta}\}. \end{aligned} \tag{4.3}$$

$$\begin{aligned} p(\gamma_i|\boldsymbol{\beta}, \boldsymbol{\delta}, \sigma^2_\gamma, \mathbf{y}, \mathbf{X}) &\propto p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \sigma^2_\gamma) \times p(\gamma_i) \\ &\propto \exp\{\sum_{v=1}^{M} Y_{iv}(\mathbf{x}_{iv}\boldsymbol{\beta} + \gamma_i + \delta_{iv})\} \times \\ &\times \exp\{-\sum_{v=1}^{M} e^{\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv}} - \frac{1}{2\sigma^2_\gamma}\gamma_i^2\}. \end{aligned} \tag{4.4}$$

$$
\begin{aligned}
P(\delta_{it}|\boldsymbol{\beta},\boldsymbol{\delta}_{-it},\sigma^2_\gamma,\sigma^2_\delta,\mathbf{y},\mathbf{X}) &\propto P(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\sigma^2_\delta)\times P(\delta_{it})\\
&\propto \exp\{Y_{it}(\mathbf{x}_{it}\boldsymbol{\beta}+\gamma_i+\delta_{it})\}\times\\
&\times \exp\{-e^{\mathbf{x}_{it}\boldsymbol{\beta}+\gamma_i+\delta_{it}}-\frac{1}{2\sigma^2_\delta}\delta^2_{it}\}. \qquad (4.5)
\end{aligned}
$$

Note que não é possível reconhecer qualquer distribuição de probabilidade a partir dos núcleos mostrados nas expressões (4.3), (4.4) e (4.5). As constantes de normalização não estão disponíveis, logo, o algoritmo Metropolis-Hastings dentro da estrutura do *Gibbs Sampling* pode ser uma alternativa na geração de valores dessas expressões. Destacamos que outros algoritmos também podem ser utilizados dentro do *Gibbs Sampling* para gerar valores candidatos de uma distribuição em que as constantes de normalização não estão disponíveis; ver Seção 3.3. Finalmente, as distribuições condicionais completas dos parâmetros $\sigma^2_\gamma$ e $\sigma^2_\delta$ apresentam forma fechada (Gama Inversa) e são dadas a seguir.

$$
\begin{aligned}
p(\sigma^2_\gamma|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\mathbf{y},\mathbf{X}) &\propto p(\boldsymbol{\gamma}|\boldsymbol{\beta},\boldsymbol{\delta},\sigma^2_\gamma,\mathbf{y},\mathbf{X})\times p(\sigma^2_\gamma)\\
&\propto \left[\prod_{i=1}^{N}(2\pi\sigma^2_\gamma)^{-\frac{1}{2}}\exp\{(-\frac{1}{2\sigma^2_\gamma}\gamma^2_i)\}\right]\times\\
&.\times \frac{g_2^{g_1}}{\Gamma(g_1)}(\sigma^2_\gamma)^{-(g_1+1)}\exp(\{-\frac{g_2}{\sigma^2_\gamma}\})\\
&\propto (\sigma^2_\gamma)^{-(\frac{N}{2}+g_1)-1}\times\exp\{-\frac{1}{\sigma^2_\gamma}(\frac{1}{2}\sum_{i=1}^{N}\gamma^2_i+g_2)\}. \qquad (4.6)
\end{aligned}
$$

Note que $\left(\sigma^2_\gamma|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\mathbf{y},\mathbf{X}\right)\sim GI\left[\frac{N}{2}+g_1,\frac{1}{2}(\sum_{i=1}^{N}\gamma^2_i+g_2)\right]$.

$$
\begin{aligned}
p(\sigma^2_\delta|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\mathbf{y},\mathbf{X}) &\propto p(\boldsymbol{\delta}|\boldsymbol{\beta},\boldsymbol{\gamma},\sigma^2_\delta,\mathbf{y},\mathbf{X})\times p(\sigma^2_\delta)\\
&\propto \left[\prod_{i=1}^{N}\prod_{v=1}^{M}(2\pi\sigma^2_\delta)^{-\frac{1}{2}}\exp\{(-\frac{1}{2\sigma^2_\delta}\delta^2_{iv})\}\right]\times\\
&\times \frac{d_2^{d_1}}{\Gamma(d_1)}(\sigma^2_\delta)^{-(d_1+1)}\exp(\{-\frac{d_2}{\sigma^2_\delta}\})\\
&\propto (\sigma^2_\delta)^{-(\frac{NM}{2}+d_1)-1}\times\exp\{-\frac{1}{\sigma^2_\delta}(\frac{1}{2}\sum_{i=1}^{N}\sum_{v=1}^{M}\delta^2_{iv}+d_2)\} \qquad (4.7)
\end{aligned}
$$

Temos aqui $(\sigma^2_\delta|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\mathbf{y},\mathbf{X})\sim GI\left[\frac{NM}{2}+d_1,\frac{1}{2}(\sum_{i=1}^{N}\sum_{v=1}^{M}\delta^2_{iv}+d_2)\right]$.

## 4.4 Conclusões do capítulo

Neste capítulo apresentamos uma breve descrição dos dados da aplicação real, que será desenvolvida no Capítulo 6, bem como o modelo e as distribuições condicionais completas necessárias. Os dados foram obtidos a partir de um estudo clínico realizado com 59 pacientes que sofrem de ataques epilépticos. Algumas estatísticas descritivas foram apresentadas e a partir das mesmas é razoável supor que há indícios de sobredispersão. Desta forma, consideramos o modelo Poisson log-linear com efeitos aleatórios como uma alternativa para modelar os dados. Lembrando que o nosso objetivo não é propor um novo modelo, mas sim usá-lo no cenário de comparação dos métodos. Finalmente, as distribuições *a priori* e as distribuições condicionais completas foram apresentadas. Vale ressaltar que não foi possível identificar algumas das condicionais completas *a posteriori* na forma fechada, sendo assim, utilizaremos o *Gibbs Sampling* com passos Metropolis-Hastings para gerar valores da conjunta *a posteriori*.

# Capítulo 5

# Estudo de simulação

Considere o modelo definido em (4.1). Seja $M = 4$ o número de visitas, $N = 59$ o número de indivíduos. Os dados são gerados com a seguinte configuração de parâmetros: $\boldsymbol{\beta} = (-0.8, 0.3, 0.8, -0.5, -0.2, 0.2)^\top$, $\sigma^2_\gamma = 0,2$ e $\sigma^2_\delta = 0,4$. Cinco covariáveis e o intercepto são utilizados nesta análise; as covariáveis são as mesmas definidas no estudo clínico com pacientes que sofrem de epilepsia, o qual foi descrito na Seção 4. Os valores reais de $\boldsymbol{\gamma}$ são gerados de uma $N_{59}(\mathbf{0}, \sigma^2_\gamma \mathbf{I}_{59})$ e $\boldsymbol{\delta}$ de uma $N_{236}(\mathbf{0}, \sigma^2_\delta \mathbf{I}_{236})$. Com isso, todos os componentes do modelo Poisson log-linear estão especificados, sendo possível calcular $\log(\mu_{iv})$ para cada $i$ e $v$ como indicado em (4.1), e gerar $y_{iv}$ de uma Poisson$(\mu_{iv})$.

Consideramos o mesmo algoritmo para gerar das três quantidades do modelo Poisson log-linear. Por exemplo, o algoritmo RWM será utilizado para gerar os efeitos fixos $\boldsymbol{\beta}$ e os efeitos aleatórios $\gamma_i$ e $\delta_{iv}$ em um único cenário dedicado ao RWM. O mesmo será feito com os demais métodos; veja a Tabela 5.1. Decidimos não misturar os métodos, pois isso levaria a muitas combinações de algoritmos, entretanto, esta será uma de nossas propostas de trabalho futuro.

Os algoritmos MCMC foram configurados de forma que todos tivessem 25 mil iterações após o período de aquecimento da cadeia (*burn-in*). Uma única cadeia de cada algoritmo foi utilizada na estimação das quantidades de interesse. O período de *burn-in* foi de 500 mil iterações para todos os algoritmos exceto para o IWLS, que precisou de 1000 iterações até convergir. Vale lembrar que o IWLS possui uma convergência mais rápida do que os outros algoritmos, pois o WLS Bayesiano fornece boas aproximações (da distribuição

| | Cenários | | | | | |
|---|---|---|---|---|---|---|
| Parâmetros | 1 | 2 | 3 | 4 | 5 | 6 |
| $\boldsymbol{\beta}$ | RWM | IWLS | AM-HA | AM-RR | RAM | VBAM |
| $\gamma_i$ | RWM | IWLS | AM-HA | AM-RR | RAM | VBAM |
| $\delta_{iv}$ | RWM | IWLS | AM-HA | AM-RR | RAM | VBAM |

Tabela 5.1: *Cenários utilizados na implementação do modelo Poisson log-linear.*

alvo) a cada iteração, de modo que rejeita-se menos a medida que as iterações avançam. Neste trabalho, não iremos utilizar a estratégia de definir um *lag* maior que 1 para seleção espaçada de valores que irão compor a amostra *a posteriori*. Estamos interessados em investigar a autocorrelação das cadeias e um $lag > 1$ iria mascarar este resultado. Os valores iniciais foram: $\boldsymbol{\beta} = \mathbf{0}_{(6\times1)}$, $\boldsymbol{\gamma} = \mathbf{0}_{(59\times1)}$, $\boldsymbol{\delta} = \mathbf{0}_{(236\times1)}$, $\sigma^2_\gamma = \sigma^2_\delta = 1$. O parâmetro de variabilidade da distribuição de propostas do RWM foi de $(0{,}00033)\mathbf{I}_6$ para o vetor $\boldsymbol{\beta}$, pois valores maiores levam a uma alta taxa de rejeição, e 1 e 4 para $\gamma_i$ e $\delta_{iv}$, respectivamente. Com essa configuração, a taxa de aceitação do RWM ficou próxima de 23,4%; as taxas de aceitação serão discutidas e apresentadas mais adiante na Tabela 5.2. Para o VBAM, foram considerados os seguintes valores iniciais: $\mathbf{m}_0 = \mathbf{0}$, $\mathbf{P}_0 = \mathbf{I}$, $\lambda_{t-1} = 2,4^2/d$ e $v_0 = d + 2$ (esses valores são sugeridos por Mbalawata et al. (2015)). As seguintes distribuições *a priori* são especificadas:

- $\boldsymbol{\beta} \sim N_6(\mathbf{0}, \mathbf{C})$ com $\mathbf{C} = \omega\mathbf{I}_{(6\times6)}$ e $\omega$ sendo uma constante positiva grande. Essa especificação implica em uma matriz de precisão $\mathbf{C}^{-1} \approx \mathbf{0}\mathbf{I}_{(6\times6)}$.

- $\sigma^2_\gamma \sim GI(g_1, g_2)$ e $\sigma^2_\delta \sim GI(d_1, d_2)$ com $g_1 = d_1 = 2{,}001$ e $g_2 = d_2 = 1{,}001$. A esperança dessas distribuições é 1 e a variância 1000. Essa especificação também é usada em Fotouhi (2008).

Para avaliar a convergência das cadeias, utilizamos as funções *geweke.diag* e *heidel.diag*, ambas do pacote *coda* (Plummer et al., 2006) e disponíveis no *software* R. O diagnóstico de convergência da função *geweke.diag* foi proposto por Geweke (1992) e é baseado na igualdade das médias de partes da cadeia. Por padrão, o teste compara as médias do trecho inicial (com 10% da cadeia) com o trecho final (com 50% da cadeia). A

estatística de teste associada a hipótese nula consiste na diferença entre as médias sobre o seu desvio padrão e, assintoticamente, tem distribuição Normal padrão. Já o diagnóstico de convergência da função *heidel.diag*, é baseado no trabalho de Heidelberger e Welch (1983) e utiliza a estatística de teste de Cramer-von-Mises para verificar se a distribuição da cadeia é estacionária. De forma resumida, sucessivas comparações são feitas entre a distribuição de toda a cadeia e as distribuições formadas por 90%, 80%, 70% e 60% da cadeia, até que a hipótese de convergência seja rejeitada ou até que reste 50% da cadeia.

As Figuras 5.1 e 5.2 exibem estimativas pontuais e intervalares para os parâmetros do modelo Poisson log-linear. A Figura 5.1 compara os algoritmos RWM, IWLS e AM-HA. Já a Figura 5.2 compara o AM-RR, RAM e VBAM. Como pode ser visto, o Painel (a) da Figura 5.1 sugere que os algoritmos apresentam resultados similares em termos de inferência; o AM-HA e o IWLS possuem intervalo de credibilidade bastante amplo para $\beta_0$. Os algoritmos RWM e AM-HA (1º e 3º gráficos no Painel (a)) indicam boas estimativas *a posteriori* para o vetor $\boldsymbol{\beta}$, exceto para $\beta_2$. Já na Figura 5.2, o Painel (a) sugere que o AM-RR tem o melhor resultado em termos de inferência e o RAM o pior. Embora o RAM e o VBAM apresentem menores intervalos de credibilidade, para a maioria dos parâmetros esses intervalos não contêm o verdadeiro valor. Veja que o AM-RR apresentou intervalo de credibilidade bastante amplo para $\beta_0$ e $\beta_3$. Nos Painéis (b) dessas figuras, estão os resultados de inferência *a posteriori* para $\sigma^2_\gamma$ e $\sigma^2_\delta$. Na Figura 5.1, note que o IWLS foi quem melhor estimou ambas as variâncias; todos os outros algoritmos superestimam tanto $\sigma^2_\gamma$ quanto $\sigma^2_\delta$. Os Painéis (c) e (d) apresentam resultados de inferência para os componentes $\gamma_i$ e $\delta_{iv}$; todos os algoritmos forneceram estimativas similares, com a grande maioria dos intervalos de 95% contendo o verdadeiro valor. Nos Painéis (c) e (d) da Figura 5.1, pode-se notar que os intervalos de credibilidade do IWLS são os menores, sugerindo menor incerteza *a posteriori*.

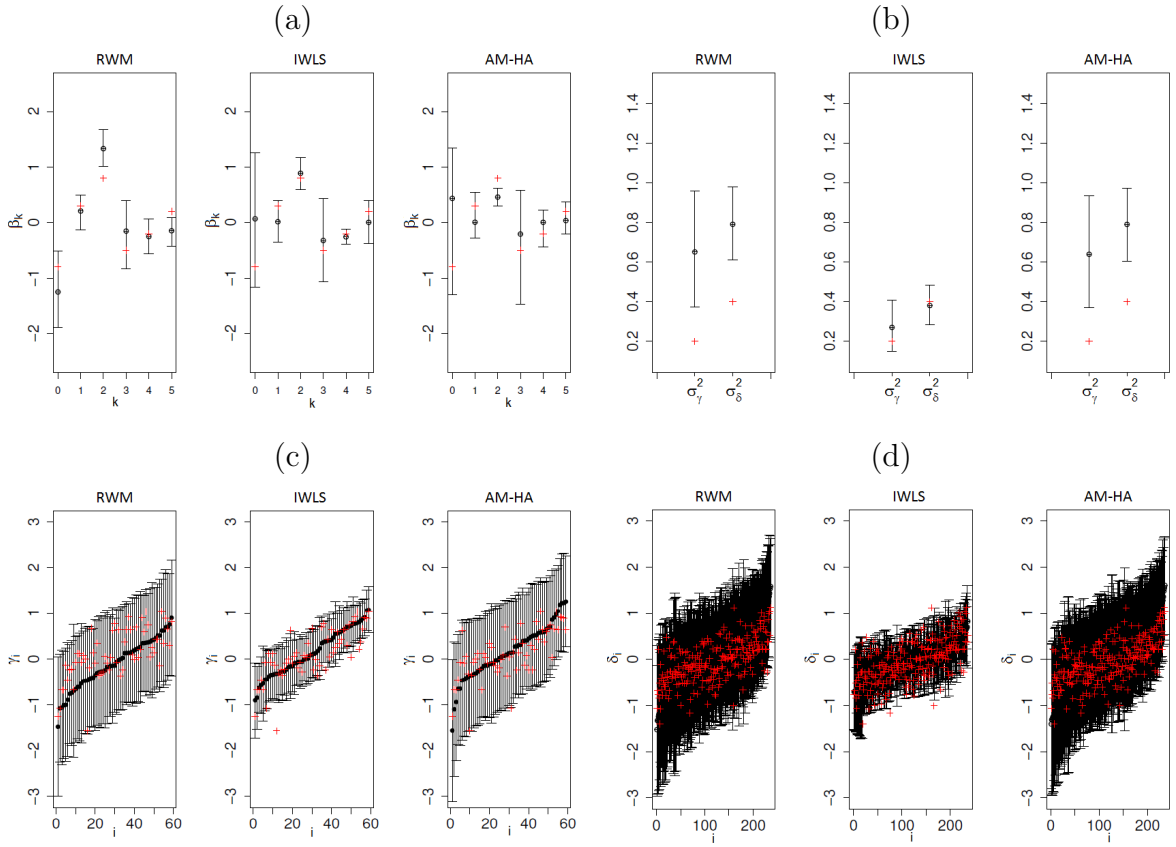


Figura 5.1: *Valores reais ( "+" vermelho), média a posteriori (círculo) e o intervalo de credibilidade HPD de 95 % (barra) para os parâmetros e efeitos aleatórios do modelo Poisson log-linear. Os algoritmos RWM, IWLS e AM-HA (nessa ordem) são comparados em cada painel. Os intervalos nos Painéis (c) e (d) estão ordenados com relação aos valores reais de* $\boldsymbol{\gamma}$ *e* $\boldsymbol{\delta}$.

O motivo pelo qual o IWLS apresentou melhor desempenho para estimar $\sigma^2_\gamma$ e $\sigma^2_\delta$ pode estar associado ao fato de suas estimativas para os efeitos $\gamma_i$ e $\delta_{iv}$ terem menor variância. Isso acontece porque as cadeias geradas pelo IWLS para $\gamma_i$ e $\delta_{iv}$ são menos autocorrelacionadas do que as cadeias geradas pelos outros algoritmos. Veja que as condicionais completas *a posteriori* de $\sigma^2_\gamma$ e $\sigma^2_\delta$, exibidas em (4.6) e (4.7), possuem forma fechada sendo distribuições Gamas Inversas com parâmetros de escala dependendo dos valores de $\gamma_i$ e $\delta_{iv}$. Ressaltamos que as cadeias de todos os parâmetros convergiram visualmente; no Apêndice C, as Figuras C.1, C.2 e a Tabela C.2 mostram as cadeias de $\sigma^2_\gamma$ e $\sigma^2_\delta$ para os algoritmos listados na Tabela 5.1 e a Estatística Z de Geweke associada

as respectivas cadeias.

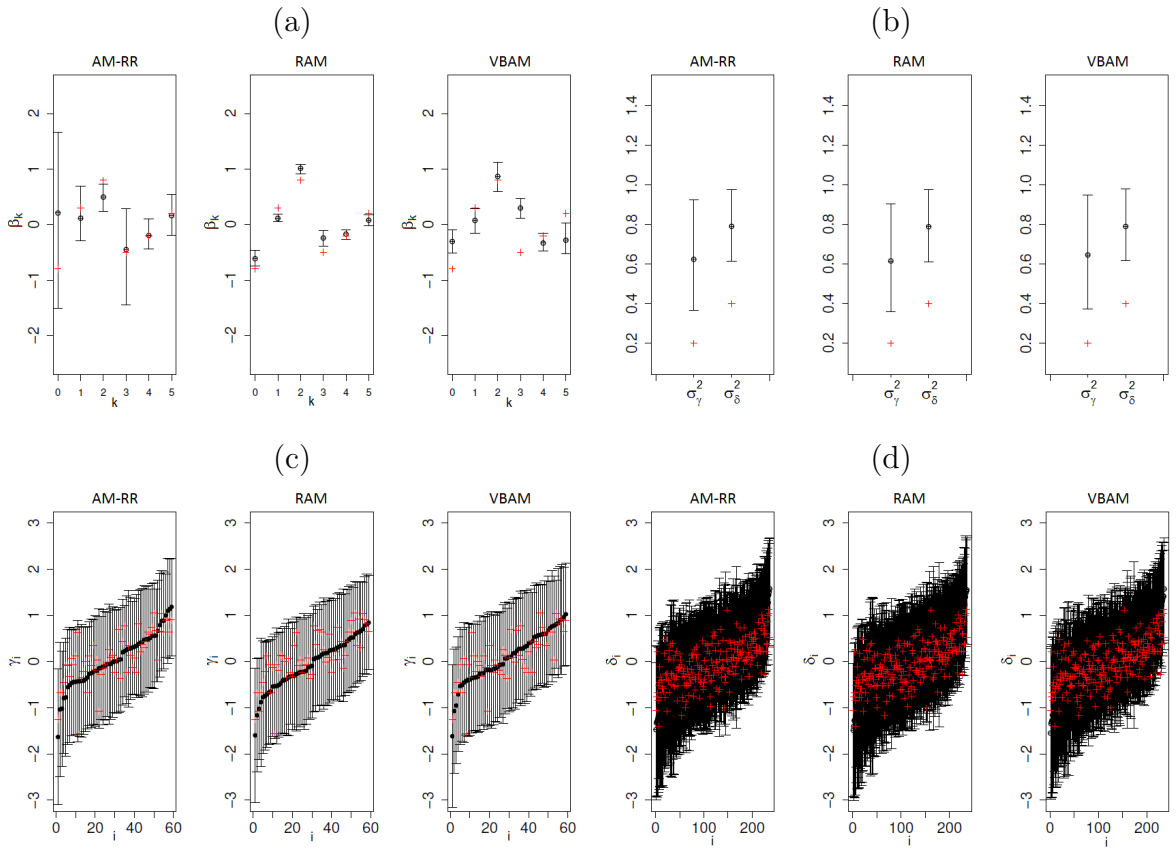


Figura 5.2: *Valores reais ("+" vermelho), média a posteriori (círculo) e o intervalo de credibilidade HPD de 95% (barra) para os parâmetros e efeitos aleatórios do modelo Poisson log-linear. Os algoritmos AM-RR, RAM e VBAM (nessa ordem) são comparados em cada painel. Os intervalos nos Painéis (c) e (d) estão ordenados em ordem crescente com relação aos valores reais de* $\boldsymbol{\gamma}$ *e* $\boldsymbol{\delta}$.

Agora, considere a expressão (3.13) para calcularmos o tamanho efetivo da amostra. Essa estatística, a qual foi descrita na Seção 3.9, será utilizada para compararmos a qualidade das estimativas dos métodos com base na autocorrelação. Quanto maior for a estatística $n_{ef}$, melhor é o *mixing* da cadeia gerada pelo algoritmo.

A Figura 5.3 compara os algoritmos em relação ao tamanho efetivo da amostra. Note que no Painel (a), o $n_{ef}$ para todo o vetor $\boldsymbol{\beta}$ é maior no IWLS; os maiores tamanhos efetivos da amostra estão associados às cadeias de $\beta_5$ e $\beta_0$. Ainda no Painel (a), note

que parece não haver diferença significativa entre os algoritmos RWM, AM-HA, AM-RR, RAM e VBAM. Devido a escala do gráfico, a altura das barras é praticamente a mesma, mas os valores são diferentes; veja a Tabela C.1 no Apêndice C. O Painel (b) mostra a estatística $n_{ef}$ para $\sigma^2_\gamma$ e $\sigma^2_\delta$; estes parâmetros não exigem o Metropolis-Hastings. Os maiores $n_{ef}$ para $\sigma^2_\gamma$ são observados no RAM e AM-RR. Já para $\sigma^2_\delta$, os maiores $n_{ef}$ são obtidos no RAM e RWM.

Dado a grande quantidade de componentes nos vetores $\boldsymbol{\gamma}$ e $\boldsymbol{\delta}$, a estatística $n_{ef}$ foi calculada para apenas três de seus componentes ($\gamma_1$, $\gamma_{30}$, $\gamma_{59}$ e $\delta_1$, $\delta_{118}$, $\delta_{236}$); estes componentes foram escolhidos de forma que pudéssemos observar diferentes indivíduos em diferentes tempos. Nestes casos, é possível notar, nos Painéis (c) e (d), que o IWLS apresenta os maiores valores de tamanho efetivo da amostra. Os Painéis (c) e (d) indicam resultados similares para os algoritmos RWM, AM-HA, AM-RR, RAM e VBAM.

Ressaltamos que os algoritmos foram configurados para fornecer uma amostra *a posteriori* de tamanho 25 mil e os tamanhos efetivos da amostra estão bem abaixo desse número, sugerindo forte autocorrelação entre os valores das cadeias. No Painel (a) da Figura 5.3, os tamanhos efetivos da amostra para o vetor $\boldsymbol{\beta}$ variam de 247 a 1920. Em ambos os casos, seria necessário configurações com (aproximadamente) 2,5 milhões e 325 mil de iterações, respectivamente, no intuito de obter um tamanho efetivo da amostra próximo de 25 mil. O Painel (b) sugere que a autocorrelação da cadeia para os parâmetros $\sigma^2_\gamma$ e $\sigma^2_\delta$ também é alta, pois os valores do $n_{ef}$ variam de 1200 a 2700 para $\sigma^2_\gamma$ e de 3200 a 5200 para $\sigma^2_\delta$. Para os três componentes dos vetores $\boldsymbol{\gamma}$ e $\boldsymbol{\delta}$, a situação não é diferente; os tamanhos efetivos da amostra variam em torno de quantidades bem menores que 25 mil.

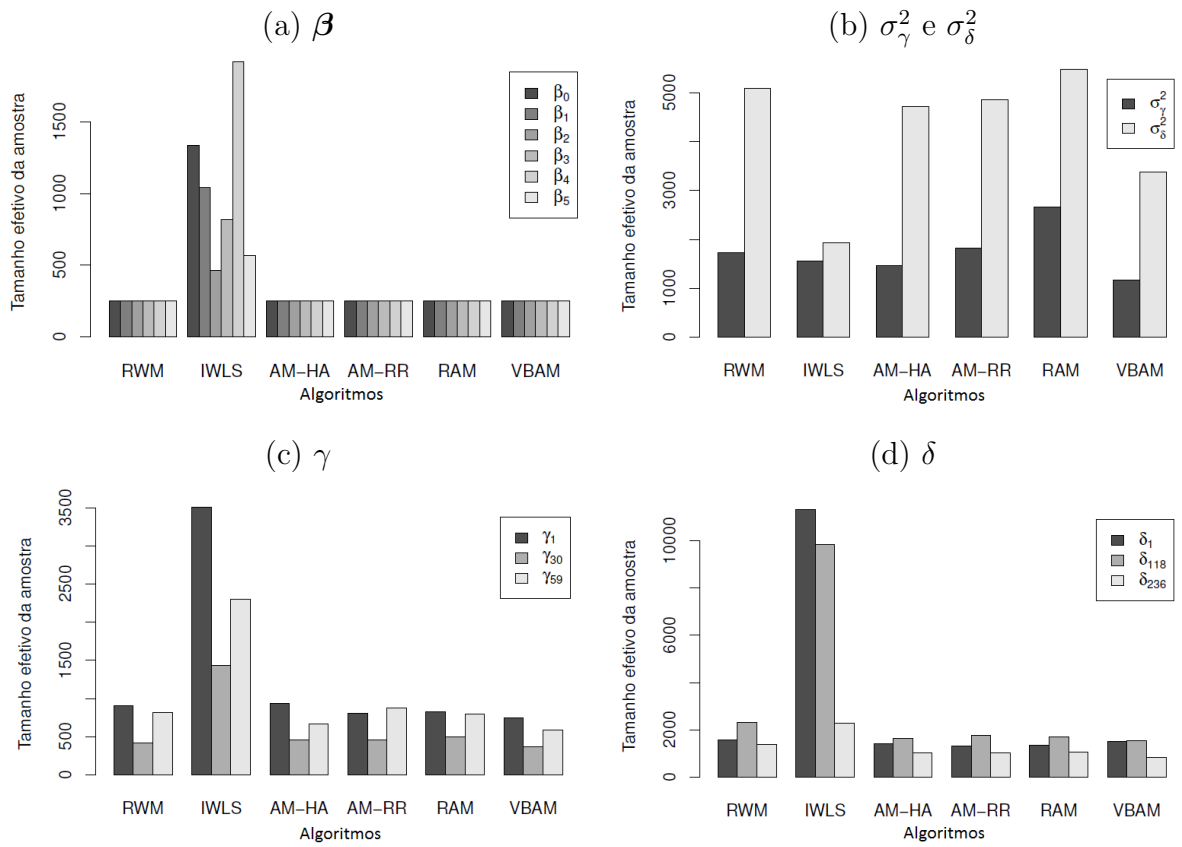


Figura 5.3: *Comparação dos algoritmos em termos de tamanho efetivo da amostra.*

Na Tabela 5.2 estão registradas as taxas de aceitação de cada algoritmo para cada componente. Para o vetor $\boldsymbol{\beta}$, as taxas variam entre 0,99% e 86,64%; os algoritmos adaptativos AM-HA e AM-RR apresentam taxas bastante baixas, que não passam de 1%; os algoritmos RWM, IWLS, RAM e VBAM apresentaram taxas de aceitação maiores, que variam de 22,88% a 86,64%. Já para os $\gamma_i$'s e $\delta_{iv}$'s, em média, as taxas ficaram acima de 73% para o IWLS, AM-HA, AM-RR e VBAM. Visto que o IWLS, AM-HA, AM-RR e VBAM são algoritmos adaptativos, não é possível configurar uma taxa de aceitação desejável ou "ideal". Por outro lado, no RWM e RAM é possível estabelecer uma taxa de aceitação desejada, que foi configurada em 23,4% neste trabalho. A constante 23,4% foi escolhida a partir dos trabalhos de Gelman et al. (1996) e Roberts et al. (1997), que mostram através de resultados teóricos e numéricos que a mesma maximiza a eficiência do RWM no caso em que as distribuições de propostas e interesse são Gaussianas. No nosso caso, a distribuição de propostas é Gaussiana e a de interesse desconhecida, porém

os histogramas de $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ e $\boldsymbol{\delta}$ indicam visualmente formato Gaussiano.

Note que para o vetor $\boldsymbol{\beta}$ a taxa de aceitação do RAM é de 40,33%, bem acima da taxa configurada. Por outro lado, para os efeitos aleatórios as taxas ficaram bastante próximas de 23,4%.

| Parâmetro | RWM | IWLS | AM-HA | AM-RR | RAM | VBAM |
|---|---|---|---|---|---|---|
| $\boldsymbol{\beta}$ | 22,88 | 50,23 | 0,99 | 1,00 | 40,33 | 86,64 |
| $\gamma_i$ | 26,07 | 80,28 | 86,77 | 87,57 | 24,00 | 88,88 |
| $\delta_{iv}$ | 26,20 | 73,24 | 94,98 | 95,54 | 22,82 | 96,32 |

Tabela 5.2: *Percentual médio da taxa de aceitação dos algoritmos RWM, IWLS, AM-HA, AM-RR, RAM e VBAM.*

Vale ressaltar que o vetor $\boldsymbol{\beta}$, em todos os algoritmos, foi atualizado como um todo através de uma distribuição normal multivariada com determinado vetor de média e matriz de covariâncias configurados de acordo com cada método. Isto é, as entradas $\beta_0$, $\beta_1$, $\beta_2$, $\beta_3$, $\beta_4$ e $\beta_5$ não foram atualizadas individualmente, mas sim conjuntamente. Devido a esse fato, as taxas de aceitação do AM-HA e AM-RR foram bastante baixas. Com isso, para esses dois algoritmos, realizamos também a atualização de cada componente de $\boldsymbol{\beta}$ separadamente. Ao fazermos isso, a taxa de aceitação média de $\boldsymbol{\beta}$, dos $\gamma_i's$ e $\delta_i's$ ficou em aproximadamente 10%, 55% e 63%, respectivamente. Em termos de estimativas e $n_{ef}$, não houveram mudanças significativas; veja a Figura C.3 no Apêndice C.

Uma vez que os métodos utilizados são diferentes em termos de complexidade e dificuldade de implementação, registramos o tempo computacional gasto por cada algoritmo em 1000 iterações utilizando a função *system.time* disponível no R. Essa informação é interessante no sentido de indicar o quão caro computacionalmente eles são. Por exemplo, o RWM não possui estrutura adaptativa e a especificação das distribuições geradoras de propostas é feita *a priori*. Em contrapartida, todos os outros métodos possuem estruturas mais elaboradas devido a atualização das matrizes de covariâncias das distribuições de propostas ao longo das iterações; o método VBAM é ainda mais complexo, pois requer um algoritmo iterativo específico para atualizar a matriz de covariâncias. A Tabela 5.3 mostra que o menor tempo foi registrado para o RWM. Os algoritmos IWLS, AM-HA e

AM-RR apresentaram tempos similares; veja que a diferença entre o AM-HA e AM-RR é de apenas um segundo. Os algoritmos RAM e VBAM foram os que demandaram mais tempo devido as suas estruturas adaptativas que requerem mais operações matriciais.

| Método | RWM | IWLS | AM-HA | AM-RR | RAM | VBAM |
|---|---|---|---|---|---|---|
| Tempo | 16 | 24 | 29 | 30 | 63 | 330 |

Tabela 5.3: *Tempo de execução (em segundos) em 1000 iterações para cada algoritmo. Todas as simulações foram desenvolvidas em um (único) computador com processador Intel Core i5-3330 3.0 GHz (4 CPUs) com 16GB de mémoria.*

A Tabela 5.4 mostra quantas iterações foram necessárias para que a estatística $n_{ef}$ atingisse o valor 1000. Nesse caso, o $n_{ef}$ alvo foi fixado em 1000, $lag = 1$ e verificou-se quantas iterações foram necessárias para que este tamanho fosse atingido para $\boldsymbol{\beta}$. Essa análise foi realizada utilizando o ARS na estimação dos efeitos aleatórios no intuito de diminuir o tempo computacional e verificar como os algoritmos RWM, IWLS, AM-HA, AM-RR, RAM e VBAM se comportam na presença de outro algoritmo; para ver os resultados relacionados a esse cenário veja as Figuras D.1 e D.2 no Apêndice D. Em paralelo, vale lembrar que não é possível utilizar o ARS para gerar (conjuntamente) de $\boldsymbol{\beta}$ pois ele é um vetor. Escolhemos fazer essa análise para $\boldsymbol{\beta}$ porque o mesmo apresentou o menor $n_{ef}$ e, por consequência, maior autocorrelação. O cálculo do $n_{ef}$ foi feito a cada mil iterações para $\boldsymbol{\beta}$. Esta estratégia foi usada visto que calcular o $n_{ef}$ é caro computacionalmente, pois a matriz que armazena as estimativas fica maior a medida que o número de iterações aumenta. Devido a isso, alguns algoritmos podem apresentar resultados iguais em nossa análise. O número de iterações necessárias para atingir o $n_{ef}$ 1000 para o RWM está entre 100.002 e 101.001. Note que o IWLS foi o algoritmo que indicou o menor número de iterações (aproximadamente 14 mil). Os algoritmos RWM, AM-HA, AM-RR, RAM e VBAM requerem um número de iterações bem superior (acima 100 mil). Uma vez que o $n_{ef}$ é função do tamanho da cadeia e de sua autocorrelação, tem-se que o IWLS é o algoritmo com os melhores resultados em termos de *mixing*.

| Algoritmos | RWM | IWLS | AM-HA | AM-RR | RAM | VBAM |
|---|---|---|---|---|---|---|
| $\boldsymbol{\beta}$ | 101.001 | 14.001 | 101.001 | 101.001 | 101.001 | 101.001 |

Tabela 5.4: *Número de iterações necessárias para que os algoritmos RWM, AM-HA, AM-RR, RAM e VBAM atinjam um tamanho efetivo da amostra igual a 1000 para o parâmetro* $\boldsymbol{\beta}$.

Levando em consideração os resultados das Tabelas 5.4 e 5.3, temos que o IWLS é o melhor, quando consideramos tempo computacional e resultados em termos de *mixing*, pois o mesmo requer um número bem inferior de iterações para atingir o $n_{ef} = 1000$ e seu custo computacional é bastante competitivo. Isto é, embora o RWM tenha registrado 16 segundos para executar 1000 iterações, para atingir o $n_{ef} = 1000$ o mesmo precisou de aproximadamente 100 mil iterações. Em contrapartida, o IWLS executou 1000 iterações em 24 segundos, porém necessitou de apenas 14 mil iterações para atingir o $n_{ef} = 1000$.

## 5.1 Conclusões do capítulo

Neste capítulo apresentamos os resultados da comparação dos diferentes algoritmos Metropolis adaptativos e não-adaptativos. Os algoritmos RWM, AM-HA, AM-RR, RAM, VBAM e ARS são comparados através do modelo Poisson log-linear com efeitos aleatórios. A configuração do modelo e dos algoritmos é mostrada bem como os resultados obtidos em termos de estimativas pontuais (valores médio) e intervalares (intervalos de credibilidade).

Os resultados mostram que o Metropolis IWLS tem o melhor desempenho, uma vez que para a grande maioria das quantidades de interesse ($\boldsymbol{\beta}$, $\gamma_i$'s e $\delta'_{iv}s$) o verdadeiro valor está contido no intervalo HPD. Além disso, o IWLS apresentou os maiores valores da estatística tamanho efetivo da amostra para quase todos os parâmetros, e precisou de menos iterações para alcançar um $n_{ef} = 1000$. As taxas de aceitação para cada algoritmo foram calculadas e valores muito baixos foram verificados para $\boldsymbol{\beta}$ nos algoritmos AM-HA (0,99%) e AM-RR (1%). Com isso, um novo cenário foi criado para o AM-HA e AM-RR, no qual o vetor $\boldsymbol{\beta}$ foi atualizado entrada a entrada, fazendo com que as taxas de

aceitação para $\boldsymbol{\beta}$ subissem para 10% em média. O tempo computacional para executar 1000 iterações também foi registrado e mostrou que o IWLS é o mais rápido dentre os algoritmos adaptativos. A convergência das cadeias foi verificada através dos diagnóstico de convergência de Geweke e Heidel.

Salientamos que as conclusões aqui tiradas ficam restritas ao modelo Poisson log-linear com efeitos aleatórios e a dados longitudinais de contagem.

# Capítulo 6

# Aplicação a dados reais

Neste capítulo iremos explorar o algoritmo IWLS, que apresentou os melhores resultados no cenário de simulação. Consideramos novamente o problema do estudo clínico que envolve pacientes epilépticos e ajustamos o modelo Poisson log-linear assumindo que a distribuição dos efeitos aleatórios é uma mistura discreta de distribuições normais visando investigar a necessidade de uma distribuição com caudas mais pesadas.

## 6.1 Análise dos dados

Iremos explorar novamente o estudo clínico realizado com 59 pacientes epilépticos; reveja a Seção 4.1 para lembrar os detalhes sobre o banco de dados.

Assim como no cenário de simulação, assumimos que $\gamma_i$ e $\delta_{iv}$ têm distribuição $N(0, \sigma^2_\gamma)$ e $N(0, \sigma^2_\delta)$, respectivamente. As distribuições *a priori* para $\boldsymbol{\beta}$, $\sigma^2_\gamma$ e $\sigma^2_\delta$ também foram escolhidas de forma a apresentarem grande variabilidade: $\boldsymbol{\beta} \sim N_6(\mathbf{0}, 100^2\mathbf{I}_6)$, $\sigma^2_\gamma \sim GI(g_1 = 2{,}001, g_2 = 1{,}001)$ e $\sigma^2_\delta \sim GI(d_1 = 2{,}001, d_2 = 1{,}001)$. Para obter as estimativas *a posteriori* dos parâmetros, uma única cadeia foi utilizada. Descartamos as 1000 primeiras iterações como período de *burn-in* e configuramos o $lag = 50$ e 500 como sendo o tamanho da amostra *a posteriori*. Nesse caso foram necessárias, ao todo, 26 mil iterações (1000 do *burn-in* + 50×500 amostras *a posteriori*).

A Tabela 6.1 apresenta as estimativas para os parâmetros do modelo Poisson log-linear associados as covariáveis descritas na Seção 4.1. Os resultados sugerem que o tratamento

| | Algoritmo IWLS | | |
|---|---|---|---|
| Parâmetros | Média | Desvio-padrão | Intervalo HPD 95% |
| Intercepto | -1,1366 | 1,2976 | [-3,7004 ;1,2491] |
| *Baseline* | 0,8725 | 0,1442 | [0,5866 ; 1,1312] |
| Tratamento | -0,9586 | 0,4032 | [-1,6824 ;-0,1877] |
| Interação | 0,3402 | 0,1986 | [-0,0322 ; 0,7291] |
| Idade | 0,3492 | 0,3736 | [-0,3604 ; 1,0499] |
| Quarta visita | -0,0802 | 0,0886 | [-0,2469 ; 0,0853] |
| $\sigma^2_\gamma$ | 0,3088 | 0,0813 | [0,1724; 0,4625] |
| $\sigma^2_\delta$ | 0,2492 | 0,0396 | [0,0396; 0,3305] |

Tabela 6.1: *Média a posteriori, desvios-padrão e intervalos HPD de 95% dos parâmetros do modelo Poisson log-linear com efeitos aleatórios normais. Utilizando os dados reais sobre epilepsia e o algoritmo IWLS.*

com o medicamento *Progabide* reduz o número médio de ataques epilépticos. Em paralelo, os resultados evidenciam que quanto maior o número de ataques epilépticos registrados no período de 8 semanas que antecedeu o início do estudo, maior é o número médio de convulsões. A interação entre as variáveis tratamento e *baseline* não foi significativa, sugerindo que o medicamento *Progabide* não tem seu efeito reduzido a medida em que a variável *baseline* aumenta. Ou seja, o medicamento *Progabide* quando na presença de um histórico de muitos ataques epilépticos, é efetivo na redução do número médio de convulsões. Os resultados obtidos são bastante similares àqueles encontrados nos trabalhos de Thall e Vail (1990), Gamerman (1997) e Komárek e Laseffre (2008).

Abaixo, a Figura 6.1 mostra os quantis observados contra os quantis teóricos da distribuição dos $\gamma_i$'s e $\delta_{iv}$'s. É possível notar um afastamento nas caudas das distribuições, sugerindo que as mesmas possuem caudas mais pesadas do que a distribuição Normal, inicialmente considerada na análise. Essa observação também foi feita por Gamerman (1997), o qual sugeriu para o problema uma alternativa baseada em mistura escala de distribuições normais. Em paralelo, Komárek e Laseffre (2008) também mostram que a abordagem de mistura Gaussiana penalizada (*penalized Gaussian mixture* - PGM), proposta por Eilers e Marx (1996), pode ser uma alternativa para lidar com a não-

normalidade da distribuição de efeitos aleatórios em situações que envolvem GLMM. Além disso, através de simulações e aplicações a dados reais, Komárek e Laseffre (2008) avaliam também o impacto que a má especificação da distribuição dos efeitos aleatórios pode ter sobre as estimativas dos coeficientes da regressão.

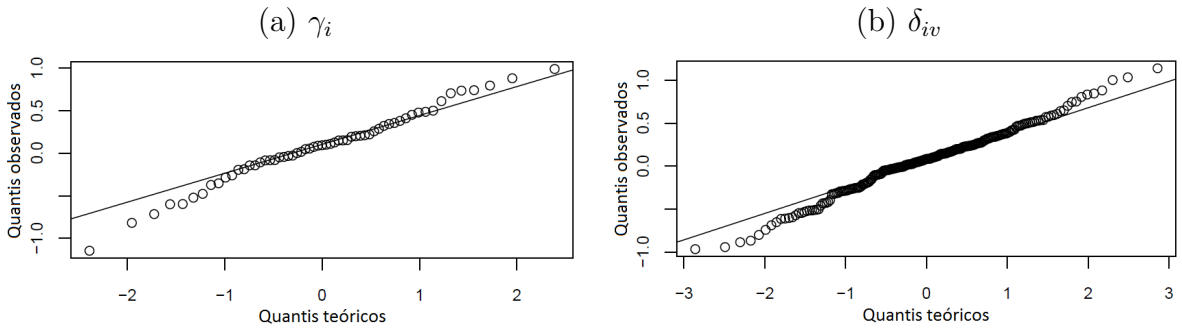


Figura 6.1: *Gráfico quantil-quantil para normalidade dos efeitos aleatórios.*

## 6.2 Modelo Poisson log-linear com mistura discreta de distribuições normais

Em situações onde os efeitos fixos ou os efeitos aleatórios de (3.6) apresentam distribuição multimodal com ou sem caldas pesadas, a distribuição Normal pode não ser razoável como distribuição. Nesse sentido, uma das alternativas que Gamerman (1997) propõe é o uso de mistura escala de distribuições normais. Por exemplo, suponhamos que é razoável assumir como distribuição para a quantidade de interesse $\gamma_i$ ou $\delta_{iv}$ a distribuição t-Student, Normal contaminada, Slash, Laplace ou Logística. A ideia utilizada por Gamerman (1997) é gerar valores dessas distribuições a partir da razão entre $B/F$, duas variáveis aleatórias independentes, sendo $B$ normal padrão e $F$ positiva. Por exemplo, a distribuição t-Student com $\nu$ graus de liberdade pode ser representada como uma razão entre $B/F$, sendo que $F = \sqrt{F^*/\nu}$ com $F^* \sim \chi^2_\nu$. Por sua vez, uma variável aleatória que tem distribuição Laplace com média zero pode ser obtida da mesma forma, sendo $F = 1/\sqrt{F^*}$ com $F^*$ tendo distribuição Exponencial. Essa representação é bastante útil quando há interesse em gerar valores de distribuições simétricas, que podem ser obtidas a partir da razão entre uma variável normal padrão e uma outra variável aleatória positiva.

Nos trabalhos de Andrews e Mallows (1974) e Lange e Sinsheimer (1993), são apresentados vários resultados relacionados às distribuições que podem ser obtidas a partir da representação mencionada acima, bem como detalhes de geração, estimação, estudos de simulação e aplicações a dados reais utilizando modelos de regressão.

Nesta seção, apresentamos uma alternativa para modelar a distribuição dos efeitos aleatórios através do uso de mistura finita de distribuições normais com médias zero e variâncias desconhecidas a serem estimadas. Com essa proposta, procuramos lidar com a não-normalidade e as caudas pesadas da distribuição dos efeitos aleatórios.

Considere a expressão (4.1) só que agora

$$\gamma_i \overset{iid}{\sim} \sum_{m=0}^{N_\gamma} \rho_{\gamma_m} N(0, \sigma^2_{\gamma_m}), \text{ com } \sum_{m=1}^{N_\gamma} \rho_{\gamma_m} = 1,$$
$$\delta_{iv} \overset{iid}{\sim} \sum_{n=0}^{N_\delta} \rho_{\delta_n} N(0, \sigma^2_{\delta_n}), \text{ com } \sum_{n=1}^{N_\delta} \rho_{\delta_n} = 1.$$

Com isso, estamos assumindo que os efeitos aleatórios seguem uma mistura de distribuições normais com média 0 e variâncias desconhecidas. Por simplicidade, e sem perda de generalidade, vamos nos restringir ao caso em que a mistura apresenta dois componentes ($N_\gamma = N_\delta = 2$). Dessa forma temos

$$\gamma_i \overset{iid}{\sim} \rho_{\gamma_0} N(0, \sigma^2_{\gamma_0}) + (1 - \rho_{\gamma_0}) N(0, \sigma^2_{\gamma_1}) \text{ e}$$
$$\delta_{iv} \overset{iid}{\sim} \rho_{\delta_0} N(0, \sigma^2_{\delta_0}) + (1 - \rho_{\delta_0}) N(0, \sigma^2_{\delta_1}).$$

Por uma questão de identificabilidade do modelo, assumimos que $\sigma^2_{\gamma_0} < \sigma^2_{\gamma_1}$ e $\sigma^2_{\delta_0} < \sigma^2_{\delta_1}$. Em paralelo, no intuito de evitar que as variâncias dos componentes da mistura apresentassem valores próximos, configuramos o algoritmo de forma que $\sigma^2_{\gamma_0}$ e $\sigma^2_{\delta_0}$ só seriam aceitas se $\sigma^2_{\gamma_0}/\sigma^2_{\gamma_1} > 0{,}1$ e $\sigma^2_{\delta_0}/\sigma^2_{\delta_1} > 0{,}1$; valores acima de 0,1 levaram a uma alta de rejeição. Além disso, criamos as variáveis $W_i$ e $Q_{iv}$, que assumem valor 1 com probabilidade $\rho_{\gamma_0}$ e $\rho_{\delta_0}$, respectivamente, para indicar de qual componente da mistura o efeito aleatório é proveniente; isto é, $W_i \sim \text{Bernoulli}(\rho_{\gamma_0})$ e $Q_{iv} \sim \text{Bernoulli}(\rho_{\delta_0})$.

Abaixo, consideramos as seguintes distribuições *a priori* para o novo modelo que assume mistura de normais para modelar os efeitos aleatórios:

- $\boldsymbol{\beta} \sim N_6(\mathbf{0}, \mathbf{C})$ com $\mathbf{C} = \omega \mathbf{I}_{(6\times 6)}$ e $\omega$ sendo uma constante positiva grande. Essa especificação implica em uma matriz de precisão $\mathbf{C}^{-1} \approx \mathbf{0}\mathbf{I}_{(6\times 6)}$.

- $\sigma^2_{\gamma_1} \sim GI(g_{1_1}, g_{2_1})$ e $\sigma^2_{\delta_1} \sim GI(d_{1_1}, d_{2_1})$ com $g_{1_1} = d_{1_1} = 2{,}001$ e $g_{2_1} = d_{2_1} = 1{,}001$. Temos aqui esperança 1 e variância 1000.

- $\sigma^2_{\gamma_0} \sim GI(g_{1_0}, g_{2_0})$ e $\sigma^2_{\delta_0} \sim GI(d_{1_0}, d_{2_0})$ com $g_{1_0} = d_{1_0} = 2{,}001$ e $g_{2_0} = d_{2_0} = 0{,}5$. Com isso, temos esperança 0,5 e variância 249,5. Essas distribuições *a priori* foram escolhidas de forma que, em média, apresentem menor variância do que $\sigma^2_{\gamma_1}$ e $\sigma^2_{\delta_1}$, respectivamente.

- $\rho_{\gamma_0} \sim \text{Beta}(a_1 = 1, b_1 = 1)$ e $\rho_{\delta_0} \sim \text{Beta}(c_1 = 1, d_1 = 1)$, ou seja, temos a U[0,1].

A partir dessa formulação, não é possível obter analiticamente a distribuição conjunta *a posteriori* de $(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \sigma^2_{\gamma_0}, \sigma^2_{\gamma_1}, \sigma^2_{\delta_0}, \sigma^2_{\delta_1})$. Como alternativa, podemos encontrar as distribuições condicionais completas e usar o *Gibbs Sampling* com passos do Metropolis IWLS. De forma geral, as distribuições condicionais completas do modelo Poisson log-linear assumindo mistura de normais para os efeitos aleatórios são similares àquelas apresentadas na Seção 4.3. A função de verossimilhança e a distribuição condicional completa para $\boldsymbol{\beta}$ permanecem iguais as expressões (4.2) e (4.3), respectivamente. Já as distribuições condicionais completas para as variâncias dos efeitos aleatórios, além de $\gamma_i$, $\delta_{iv}$, $W_i$, $Q_{iv}$, $\rho_{\gamma_0}$ e $\rho_{\delta_0}$ são apresentadas no Apêndice E.

A Tabela 6.2 apresenta os resultados do modelo Poisson log-linear assumindo mistura de normais como distribuição dos efeitos aleatórios. Note que as estimativas dos parâmetros não mudaram muito com relação àquelas apresentadas na Tabela 6.1, quando na ocasião assumimos que os efeitos aleatórios eram normalmente distribuídos. Os resultados exibidos aqui também são similares àqueles apresentados em Komárek e Laseffre (2008), os quais assumiram que a distribuição dos efeitos aleatórios é, para o mesmo modelo e problema, uma PGM e um processo de Dirichlet. Aqui, assumimos que a distribuição dos efeitos $\gamma_i$ e $\delta_{iv}$ é uma mistura de duas distribuições normais com média zero e variâncias a serem estimadas.

| | Algoritmo IWLS | | |
|---|---|---|---|
| Parâmetros | Média | Desvio-padrão | Intervalo HPD 95% |
| Intercepto | -1,2884 | 1,0766 | [-3,3325; 0,7266] |
| *Baseline* | 0,8674 | 0,14365 | [0,5903; 1,1378] |
| Tratamento | -1,0059 | 0,4360 | [-1,9086; -0,1599] |
| Interação | 0,3666 | 0,2275 | [-0,0784; 0,8284] |
| Idade | 0,4044 | 0,3195 | [-0,1710; 1,0438] |
| Quarta visita | -0,0825 | 0,0913 | [-0,2480; 0,0887] |
| $\sigma^2_{\gamma_0}$ | 0,2192 | 0,0677 | [0,0907; 0,3411] |
| $\sigma^2_{\gamma_1}$ | 0,3623 | 0,1064 | [0,1959; 0,5928] |
| $\sigma^2_{\delta_0}$ | 0,2260 | 0,0378 | [0,1549; 0,2954] |
| $\sigma^2_{\delta_1}$ | 0,2955 | 0,0517 | [0,2091; 0,3908] |

Tabela 6.2: *Média a posteriori, desvios-padrão e intervalos HPD de 95% dos parâmetros do modelo Poisson log-linear com efeitos aleatórios mistura de normais. Utilizando os dados reais sobre epilepsia e o algoritmo IWLS.*

Na Figura 6.2 são apresentados o gráfico quantil-quantil e o da densidade dos efeitos aleatórios, agora considerando o modelo com mistura. No Painel (a), é possível perceber que apenas dois valores $\gamma_i$ se distaciam com maior destaque da linha "teórica", sendo eles relacionados aos indivíduos 35 (no canto inferior esquerdo) e 58 (no canto superior direito), os quais apresentaram um número médio de ataques epilépticos ao final das visitas igual a 0 e 18,5, respectivamente. Ambos os pacientes receberam o medicamento *Progabide* porém o paciente 35, que no início tinha uma média de 8 convulsões a cada duas semanas, ao final do estudo registrou um número médio de 18,5 convulsões. Já o paciente 58, que tinha um histórico de 3 convulsões a cada duas semanas, reduziu a zero os ataques epilépticos durante o estudo. No painel (b), são apresentadas as densidades estimadas dos efeitos aleatórios $\delta_{iv}$ considerando o modelo com e sem mistura. Nota-se que as densidades estão bem próximas, sugerindo que a mistura de normais não é tão vantajosa para esta análise de dados visto que ela não indica uma cauda mais pesada do que a da Normal.

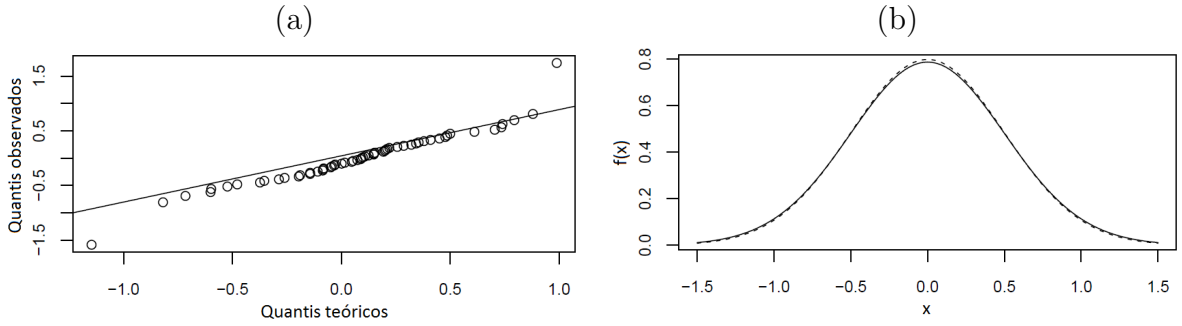


Figura 6.2: *Gráfico dos efeitos aleatórios: (a) Quantis teóricos versus quantis observados de $\gamma_i$'s, (b) densidades estimadas dos efeitos aleatórios $\delta_{iv}$: modelo com mistura (linha tracejada), modelo sem mistura (linha contínua).*

No intuito de selecionar o melhor modelo, podemos utilizar alguns critérios de comparação como, por exemplo, o *Deviance Information Criterion* - DIC, *Watanabe-Akaike Information Criterion* - WAIC e o *Log-Pseudo Marginal Likelihood* - LPML. O DIC (Spiegelhalter et al., 2002) consiste na diferença entre o logaritmo da função de verossimilhança, dado a média das estimativas *a posteriori* (das quantidades de interesse $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ e $\boldsymbol{\delta}$), menos o número efetivo de parâmetros do modelo. Isto é,

$$\text{DIC} = 2\left(\log p(\mathbf{y}|\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\delta}}) - \frac{4}{T}\sum_{t=1}^{T} \log p(\mathbf{y}|\boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\delta}^{(t)})\right). \tag{6.1}$$

O WAIC, proposto por Watanabe (2002), envolve a diferença entre a soma, para todos os indivíduos, do valor esperado do logaritmo da p.d.f ou p.m.f com relação as quantidades de interesse, menos duas vezes a soma das diferenças entre o logaritmo do valor esperado da p.d.f ou p.m.f e o valor esperado do logaritmo da função de verossimilhança, ambos com relação as quantidades de interesse.

$$\text{WAIC} = -\sum_{i=1}^{N} \log\left(\frac{1}{T}\sum_{t=1}^{T} p(Y_i|\boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\delta}^{(t)})\right) - 2\text{C}, \tag{6.2}$$

com $\text{C} = \sum_{i=1}^{N}\left(\log\left(\frac{1}{T}\sum_{t=1}^{T} p(Y_i|\boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\delta}^{(t)})\right) - \frac{1}{T}\sum_{t=1}^{T}\log p(\mathbf{y}|\boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\delta}^{(t)})\right)$. Tanto para o DIC quanto para o WAIC, menores valores implicam em um melhor modelo. Já para o LPML, que consiste na soma do logaritmo do valor esperado do inverso da p.d.f

ou p.m.f associada a variável resposta $Y_i$, quanto maior seu valor, melhor é o modelo.

$$\text{LPML} = \sum_{i=1}^{N} \log \left( \frac{1}{T} \sum_{t=1}^{T} p(Y_i|\boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\delta}^{(t)})^{-1} \right). \tag{6.3}$$

Em Chen et al. (2000) e Gelman et al. (2014) outros critérios são apresentados, bem como mais detalhes sobre os critérios mencionados acima.

Na Tabela 6.3, são apresentados os valores dos critérios de comparação DIC, WAIC e LPML para o modelo Poisson log-linear com e sem mistura de distribuições normais para os efeitos aleatórios. Note que de acordo com o DIC, parece não haver diferença significativa entre os modelos, uma vez que os valores são bastante similares. Agora, considerando o WAIC, temos que o modelo com mistura de normais apresentou um melhor ajuste aos dados em relação ao caso sem mistura. Por outro lado, se observamos o LPML, temos que o modelo sem mistura é melhor do que o com mistura. Sumarizando, temos um critério que sugere não haver diferença entre os modelos (DIC), outro que fornece evidências a favor do modelo com mistura (WAIC) e outro que indica o modelo sem mistura (LPML). Portanto, não é possível afirmar para esta aplicação que o modelo com mistura apresenta um ajuste melhor do que o modelo sem mistura.

| | Modelo | |
|---|---|---|
| Critério | Sem mistura | Com mistura |
| DIC | -11.802,94 | -11.802,15 |
| WAIC | -10.272,15 | -10.330,4 |
| LPML | 5.093,04 | 4.899,45 |

Tabela 6.3: *Critérios de comparação DIC, WAIC e LPML para os modelos Poisson log-linear com e sem mistura de distribuições normais para os efeitos aleatórios.*

## 6.3 Conclusões do capítulo

Neste capítulo, apresentamos os resultados do ajuste do modelo Poisson log-linear para os dados do estudo clínico com pacientes epilépticos descrito no Capítulo 4. Os resultados sugerem que a covariável "Tratamento" (indicando o uso do medicamento) é

estatisticamente significativa e que não existe interação entre a eficácia do medicamento e o histórico de convulsões. Isto é, o medicamento é eficaz tanto na presença de um baixo ou de um alto número de ataques. Além disso, observamos através de gráficos quantil-quantil que a distribuição dos efeitos aleatórios parece possuir caudas mais pesadas do que a distribuição Normal.

Sugerimos uma modelagem com mistura discreta de distribuições normais na tentativa de lidar com a questão das caudas pesadas da distribuição dos efeitos aleatórios. Os resultados do modelo com mistura são bastante similares àqueles do caso sem mistura e aos obtidos por Komárek e Laseffre (2008). Os critérios de comparação DIC, WAIC e LPML foram usados para comparar os modelos, porém eles divergem na indicação do melhor, sugerindo que preocupação com as caudas pesadas pode ser irrelevante nesta aplicação real. Vale ressaltar que Komárek e Laseffre (2008) também avaliaram o impacto da má especificação da distribuição dos efeitos aleatórios para esse problema, porém assumindo uma PGM e um processo de Dirichlet. O trabalho nesta referência, também conclui que parece não haver impacto significativo das diferentes especificações da distribuição dos efeitos aleatórios sobre as estimativas dos efeitos fixos.

# Capítulo 7

# Conclusões

Os métodos MCMC são importantes ferramentas de simulação especialmente em situações onde a integração usual é bastante complicada e, no contexto de inferência Bayesiana, essas situações aparecem quando queremos gerar amostras de uma distribuição *a posteriori* que não possui forma fechada ou tem alta dimensionalidade. Os algoritmos MCMC geram valores com uma estrutura de dependência Markoviana e que, por consequência, não são independentes. Com isso, algumas propriedades são requeridas ao algoritmo de forma a garantir a convergência da cadeia gerada, são elas: estacionariedade, irredutibilidade, recorrência e aperiodicidade.

Este trabalho compara seis diferentes estratégias para gerar valores candidatos no algoritmo Metropolis-Hastings. As estratégias consideradas foram: o tradicional e muito utilizado na literatura Metropolis *Random Walk*, uma versão do algoritmo WLS desenvolvido para o contexto de GLMM, o qual chamamos de Metropolis IWLS, e algoritmos Metropolis adaptativos. O algoritmo Metropolis *Random Walk* é um método MCMC bastante popular e que fornece bons resultados se a especificação da matriz de covariâncias da distribuição de propostas for feita corretamente, porém em determinadas situações essa especificação não é trivial. Algoritmos adaptativos como IWLS, AM, RAM e VBAM foram propostos como alternativas ao Metropolis *Random Walk*, uma vez que lidam com a especificação da matriz de covariâncias de forma adaptativa. Outros algoritmos, que pode ser utilizados na geração de amostra de uma distribuição, como ARS, ARMS e *Slice Sampling*, embora adaptativos, não requerem especificação de uma matriz de covariâncias

da distribuição de propostas.

Para efeitos comparativos, um modelo Poisson log-linear com efeitos aleatórios é escolhido. Esse modelo foi considerado devido a natureza dos dados de contagem do estudo clínico com pacientes epilépticos, e por permitir o tratamento da sobredispersão através da introdução de efeitos aleatórios. Vale ressaltar que seria possível utilizar outros modelos que foram apresentados, por exemplo, por Booth et al. (2003), os quais estenderam o modelo Binomial negativo log-linear para modelar a dependência das contagens e dos efeitos aleatórios ao longo do tempo. Em paralelo, Kleinman e Ibrahim (1998), apresentaram uma abordagem não-paramétrica para modelar a distribuição dos efeitos aleatórios em GLMM usando processo Dirichlet. Em ambos os trabalhos, são apresentadas alternativas para se lidar com a restrição da igualdade da média e da variância do modelo Poisson e com a especificação da distribuição dos efeitos aleatórios para situações de não normalidade.

Um estudo de simulação foi realizado e os resultados de inferência analisados para todos os algoritmos. As estimativas *a posteriori* sugerem melhor *performance* do IWLS para a maioria dos parâmetros. Por exemplo, o IWLS indica baixa incerteza *a posteriori* para $\boldsymbol{\gamma}$ e $\boldsymbol{\delta}$, e seu intervalo HPD para $\sigma_\gamma^2$ e $\sigma_\delta^2$ contém o valor real. Em paralelo, os algoritmos adaptativos RAM e VBAM apresentaram os piores resultados em termos de estimativas pontuais e intervalares. Já os algoritmos AM-HA e AM-RR apresentaram baixas taxas de aceitação para $\boldsymbol{\beta}$. Além disso, a estatística tamanho efetivo da amostra, $n_{ef}$, é assumida como critério de comparação, e novamente o IWLS mostrou-se melhor, especialmente para $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ e $\boldsymbol{\delta}$. O tempo computacional e a quantidade de iterações até atingir um tamanho efetivo da amostra igual 1000 também foram registrados e o IWLS foi o mais competitivo dentre todos os algoritmos considerados.

Adicionalmente, utilizamos o Metropolis IWLS em uma aplicação a dados reais envolvendo um estudo clínico realizado com pacientes epilépticos, assumindo que a distribuição dos efeitos aleatórios do modelo Poisson log-linear é Normal, em um primeiro momento, e, posteriormente, dada por uma mistura de distribuições normais com médias zero e variâncias desconhecidas. Em termos de inferência, os modelos forneceram estimativas parecidas. A partir dos critérios de comparação (DIC, WAIC e LPML), não foi possível

concluir sobre qual modelo apresentou o melhor ajuste aos dados, pois os critérios divergem quanto a isso, indicando que o modelo com mistura não foi vantajoso para esta aplicação (em relação à suposição de efeitos aleatórios normais).

Finalmente, é razoável dizer que o IWLS apresentou os melhores resultados quando comparado aos outros algoritmos. Quando comparado aos algoritmos adaptativos estudados aqui, foi o mais eficiente (para convergência das cadeias e tempo computacional) e problemas númericos não foram observados em sua utilização. No entanto, é importante destacar que as conclusões tiradas nesse trabalho são restritas a esse tipo de modelo (Poisson log-linear) e aos dados baseados no estudo real sobre a epilepsia. Outro detalhe importante é que o IWLS possui limitações quanto a distribuição da variável resposta de interesse que, necessariamente, precisa pertencer à família exponencial.

## 7.1 Trabalhos futuros

Como trabalhos futuros, pode-se investigar a *performance* dos métodos em outros tipos de GLM e/ou GLMM como, por exemplo, dados com resposta binária, politômica (ordinal ou nominal) e resposta restrita ao intervalo (0,1). Além disso, também pode-se considerar modelos de sobrevivência, como o modelo de riscos proporcionais e/ou modelos de fragilidade.

Pode-se avaliar até que ponto vale a pena, levando em consideração a dimensão da quantidade de interesse, gerar em bloco ou separadamente. Por exemplo, os algoritmos AM-HA e AM-RR apresentaram baixas taxas de aceitação para o vetor $\boldsymbol{\beta}$, que tinha dimensão 6, enquanto registraram melhores taxas de aceitação quando foram utilizados para amostrar individualmente os efeitos aleatórios $\gamma_i$ e $\delta_{iv}$.

Pode-se analisar a proposta de mistura de duas normais como distribuição dos efeitos aleatórios em outros modelos e/ou utilizando outros bancos de dados onde a suposição de normalidade não é razoável. Podemos comparar esta proposta com as sugestões de Gamerman (1997) e Komárek e Laseffre (2008).

Pode-se avaliar o impacto de transformações da quantidade de interesse no desempenho do método de geração. Por exemplo, uma transformação 1 a 1 pode ser aplicada para

reescalar os efeitos aleatórios para o intervalo (0,1) e isso permite o uso mais eficiente do algoritmo *Slice Sampling* que apresentou problemas computacionais na escala original. Visto que a transformação é 1 a 1, podemos retornar para a escala original ao final do algoritmo.

Adicionalmente, pode-se analisar a qualidade das estimativas *a posteriori* e a auto-correlação das cadeias considerando implementações que aplicam diferentes algoritmos para cada bloco de parâmetros. Por exemplo, usar o IWLS para estimar os efeitos fixos e os algoritmos adaptativos para os efeitos aleatórios e vice-versa.

# Apêndice

## Apêndice A: *Slice Sampling*

Introduzido por Neal (2003), o *Slice Sampling* (SS) é um método MCMC que também pode ser utilizado para gerar amostras de uma distribuição de interesse. Através da introdução de uma variável auxiliar que define a região de onde os valores candidatos serão provenientes, o SS gera uniformemente valores candidatos nessa região, mesmo para casos em que a distribuição de interesse não é log-côncava e/ou univariada.

Suponha que temos interesse em amostrar valores de $p(\theta_k|\boldsymbol{\theta}_{-k})$, com $\theta_k \in \Theta$. De forma resumida, o SS gera valores candidatos de $p(\theta_k|\boldsymbol{\theta}_{-k})$ da seguinte maneira:

1. Define um valor inicial $\theta_k^{(0)} \in \Theta$;

2. Amostra $u$ de $U[0, p(\theta_k^{(0)}|\boldsymbol{\theta}_{-k})]$ e;

3. Define um conjunto $\mathcal{S} = \{\theta_k : 0 < u < p(\theta_k|\boldsymbol{\theta}_{-k})\}$. A partir disso, o algoritmo encontra os limites inferior e superior de $\mathcal{S}$ e em seguida gera uniformemente um novo valor $\theta_k^{(1)}$ levando em conta $\mathcal{S}$ e;

4. Volta ao passo 2 até que uma amostra de tamanho desejado seja obtida após a convergência.

De acordo com Neal (2003), pode-se pensar em encontrar os limites inferior e superior de $\mathcal{S}$ de algumas maneiras. A primeira é estabelecendo um valor $w$, de modo a encontrar um intervalo de tamanho $w$ em torno de $\theta_k^{(0)}$ e expandi-lo em $w$ unidades até que seus limites estejam fora de $\mathcal{S}$. Uma outra maneira é estabelecer um intervalo em torno de $\theta_k^{(0)}$, de tamanho $w$, e a cada iteração, dobrar o seu tamanho até que seus limites

estejam fora de $\mathcal{S}$; para mais detalhes veja Neal (2003). No presente trabalho, tentamos utilizar o SS para fazer a estimação dos efeitos aleatórios do modelo Poisson log-linear, no entanto, tivemos problemas na geração dos valores candidatos; utilizamos a função em código R disponibilizada por Neal (2003)[1]. Nos testes que realizamos, foram estabelecidos diferentes chutes iniciais, porém, na maioria das vezes, o algoritmo demorou muito para encontrar os limites do intervalo $\mathcal{S}$. Diante destas dificuldades, optamos por não mostrar resultados baseados no *Slice Sampling* nesta dissertação.

# Apêndice B: algoritmo IWLS para o modelo Poisson log-linear.

Vetor de observações transformadas e sua respectiva matriz diagonal de pesos para $\boldsymbol{\beta}$, $\gamma_i$ e $\delta_{iv}$ do modelo log-linear de acordo com o IWLS proposto por Gamerman (1997).

- Inicie com $\boldsymbol{\beta} = \boldsymbol{\beta}^{(0)}$ e $t = 1$;
- Gere o valor candidato $\boldsymbol{\beta}^*$ da $N_d(\mathbf{m}^{(t)}, \mathbf{C}^{(t)})$ sendo

$$\begin{aligned}
\mathbf{m}^{(t)} &= [\mathbf{C}^{-1} + \mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}^{(t-1)})\mathbf{X}]^{-1}[\mathbf{C}^{-1}\mathbf{a} + \mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}^{(t-1)})\tilde{\mathbf{y}}(\boldsymbol{\beta}^{(t-1)})], \\
\mathbf{C}^{(t)} &= [\mathbf{C}^{-1} + \mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}^{(t-1)})\mathbf{X}]^{-1}, \\
\mathbf{W}(\boldsymbol{\beta}) &= \text{diag}[W_{1,1}(\boldsymbol{\beta}), ..., W_{N,M}(\boldsymbol{\beta})], \\
W_{iv}(\boldsymbol{\beta}) &= \exp\{\mathbf{x}_{iv}\boldsymbol{\beta} + \gamma_i + \delta_{iv}\}, \\
\tilde{\mathbf{y}}(\boldsymbol{\beta}) &= (\tilde{y}_{1,1}(\boldsymbol{\beta}), ..., \tilde{y}_{N,M}(\boldsymbol{\beta}))^\top, \\
\tilde{y}_{iv}(\boldsymbol{\beta}) &= \mathbf{x}_{iv}\boldsymbol{\beta} + \frac{Y_{iv}}{e^{\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv}}} - 1.
\end{aligned}$$

Similarmente, assumindo que $\gamma_i \sim N(0, \sigma^2_\gamma)$, temos para o efeito aleatório $\gamma_i$:

- Inicie com $\gamma_i = \gamma_i^{(0)}$ e $t = 1$;
- Gere o valor candidato $\gamma_i^*$ da $N(m_i^{(t)}, C_i^{(t)})$ sendo

[1]Disponível em: http://www.cs.toronto.edu/∼radford/ftp/slice-R-prog. Acessada em 19/09/2015

$$
\begin{aligned}
m_i^{(t)} &= \left[\frac{1}{\sigma_\gamma^2} + \mathbf{z}_i^\top \mathbf{W}_i(\gamma_i^{(t-1)})\mathbf{z}_i\right]^{-1} \left[\mathbf{z}_i^\top \mathbf{W}_i(\gamma_i^{(t-1)})\tilde{\mathbf{y}}_i(\gamma_i^{(t-1)})\right], \\
C_i^{(t)} &= \left[\frac{1}{\sigma_\gamma^2} + \mathbf{z}_i^\top \mathbf{W}_i(\gamma_i^{(t-1)})\mathbf{z}_i\right]^{-1}, \\
\mathbf{z}_i &= \text{vetor coluna (M-dimensional) preenchido com 1;} \\
\mathbf{W}_i(\gamma_i) &= \text{diag}[W_{i,1}(\gamma_i), ..., W_{i,M}(\gamma_i)], \\
W_{iv}(\gamma_i) &= \exp\{\mathbf{x}_{iv}\beta + \gamma_i + \delta_{iv}\}, \\
\tilde{\mathbf{y}}_i(\gamma_i) &= (\tilde{y}_{i,1}(\gamma_i), ..., \tilde{y}_{i,M}(\gamma_i))^\top, \\
\tilde{y}_{iv}(\gamma_i) &= \gamma_i + \frac{Y_{iv}}{e^{\mathbf{x}_{iv}\beta+\gamma_i+\delta_{iv}}} - 1.
\end{aligned}
$$

De forma análoga, assumindo que $\delta_{iv} \sim N(0, \sigma_\delta^2)$, temos para o efeito aleatório $\delta_{iv}$ os seguintes passos:

- Inicie com $\delta_{iv} = \delta_{iv}^{(0)}$ e $t = 1$;
- Gere o valor candidato $\delta_{iv}^*$ da $N(m_{iv}^{(t)}, C_{iv}^{(t)})$ sendo

$$
\begin{aligned}
m_{iv}^{(t)} &= \left[\frac{1}{\sigma_\delta^2} + W_{iv}(\delta_{iv}^{(t-1)})\right]^{-1} \left[W_{iv}(\delta_{iv}^{(t-1)})\tilde{y}_{iv}(\delta_{iv}^{(t-1)})\right], \\
C_{iv}^{(t)} &= \left[\frac{1}{\sigma_\delta^2} + W_{iv}(\delta_{iv}^{(t-1)})\right]^{-1}, \\
W_{iv}(\delta_{iv}) &= \exp\{\mathbf{x}_{iv}\boldsymbol{\beta} + \gamma_i + \delta_{iv}\}, \\
\tilde{y}_{iv}(\delta_{iv}) &= \delta_{iv} + \frac{Y_{iv}}{e^{\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv}}} - 1.
\end{aligned}
$$

# Apêndice C: Tabelas e gráficos extras

Apresentamos aqui algumas tabelas e gráficos extras complementando as análises desenvolvidas no Capítulo 5.

| | Parâmetros | | | | | |
|---|---|---|---|---|---|---|
| Métodos | $\beta_0$ | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\beta_4$ | $\beta_5$ |
| RWM | 249,71 | 248,74 | 249,04 | 249,37 | 252,49 | 250,18 |
| IWLS | 1339,35 | 1041,44 | 464,96 | 818,73 | 1920,29 | 568,80 |
| AM-HA | 248,46 | 248,36 | 248,60 | 248,39 | 247,97 | 249,03 |
| AM-RR | 251,98 | 250,90 | 250,74 | 250,23 | 251,73 | 250,43 |
| RAM | 248,40 | 249,71 | 248,99 | 248,12 | 249,54 | 248,68 |
| VBAM | 248,66 | 249,13 | 248,73 | 248,74 | 249,37 | 248,26 |

Tabela C.1: *Valores do $n_{ef}$ dos coeficientes da Figura 5.3 (Painel (a)).*

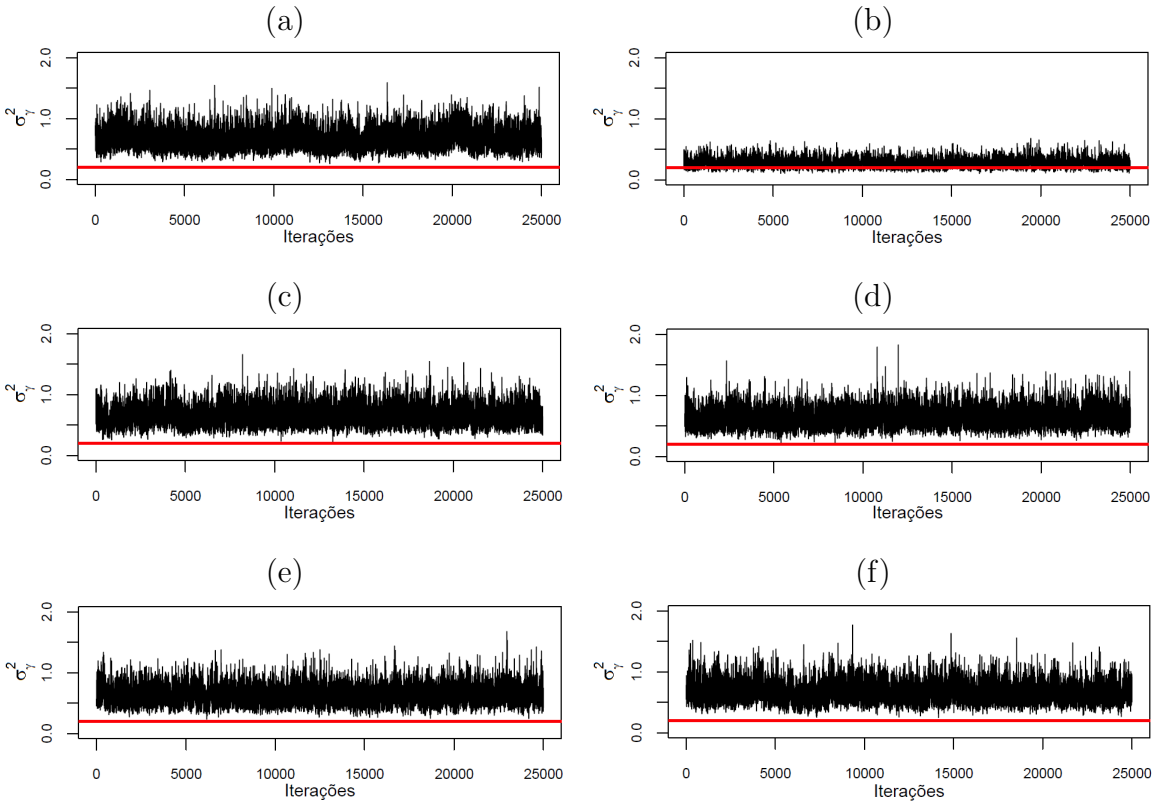


Figura C.1: *Cadeia da variância $\sigma^2_\gamma$. Algoritmos RWM (a), IWLS (b), AM-HA (c), AM-RR (d), RAM (e) e VBAM (f). A linha horizontal indica o valor real (0.2).*

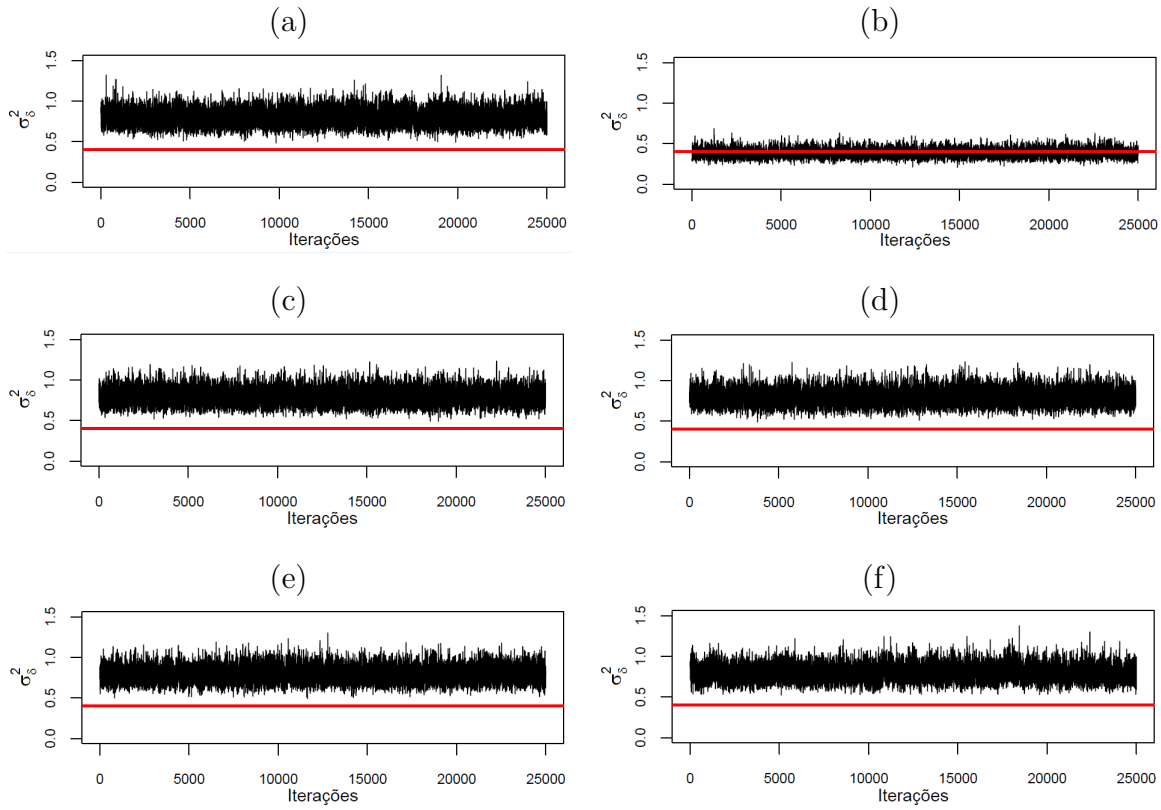


Figura C.2: *Cadeia da variância $\sigma^2_\delta$. Algoritmos RWM (a), IWLS (b), AM-HA (c), AM-RR (d), RAM (e) e VBAM (f). A linha horizontal indica o valor real (0.4).*

| | Métodos | | | | | |
|---|---|---|---|---|---|---|
| Parâmetros | RWM | IWLS | AM-HA | AM-RR | RAM | VBAM |
| $\sigma^2_\gamma$ | 1,8222 | -0,0798 | -2,7691 | -2,3943 | -0,6946 | 1,7746 |
| $\sigma^2_\delta$ | -0,6112 | -1,2787 | 0,7363 | -2,6937 | 0,0084 | -2,3039 |

Tabela C.2: Estatística Z do teste de Geweke para as cadeias das Figuras C.1 e C.2.

Note que os valores Z da Tabela C.2, referentes as cadeias das Figuras C.1 e C.2, estão entre -2,77 e 2,77, evidenciando que as cadeias de $\sigma^2_\gamma$ e $\sigma^2_\delta$ convergeriam ao nível de 99% de confiança. Já ao nível de 95% de confiança (-1,95 e 1,95), temos apenas algumas delas convergindo, porém elas são estáveis e mostram convergência na análise visual.

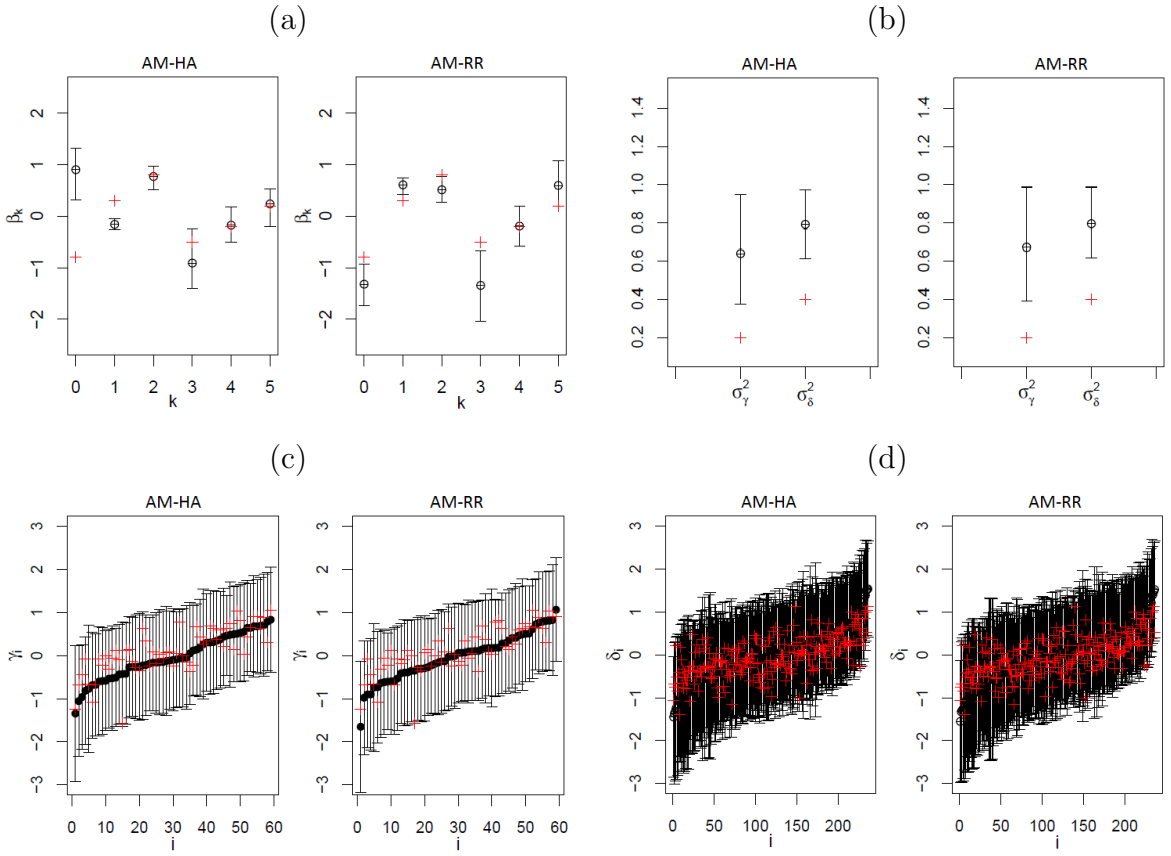


Figura C.3: *Cenário que atualiza as entradas de* $\boldsymbol{\beta}$ *individualmente. Valores reais ("+" vermelho), média a posteriori (círculo) e o intervalo HPD de 95 % (barra) para os parâmetros do modelo Poisson log-linear. Os algoritmos AM-HA e AM-RR (nessa ordem) são comparados em cada painel. Os intervalos nos Painéis (c) e (d) estão ordenados em ordem crescente com relação aos valores reais de* $\boldsymbol{\gamma}$ *e* $\boldsymbol{\delta}$.

# Apêndice D: Gráficos extras do estudo envolvendo o ARS

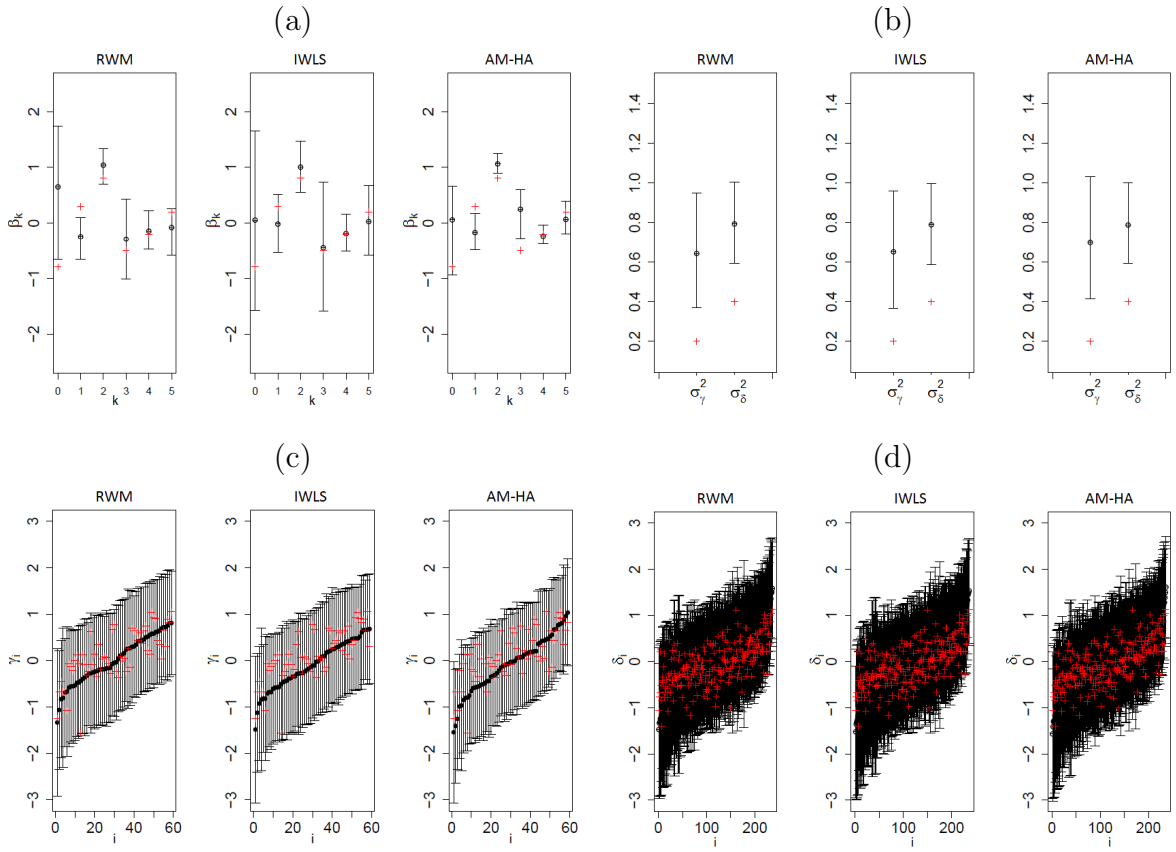


Figura D.1: *Cenário utilizando o ARS na estimação dos efeitos aleatórios $\gamma_i$ e $\delta_{iv}$. Valores reais ("+" vermelho), média a posteriori (círculo) e o intervalo HPD de 95 % (barra) para os parâmetros do modelo Poisson log-linear. Os algoritmos RWM, IWLS e AM-HA (nessa ordem) são comparados em cada painel. Os intervalos nos Painéis (c) e (d) estão ordenados em ordem crescente com relação aos valores reais de $\boldsymbol{\gamma}$ e $\boldsymbol{\delta}$.*

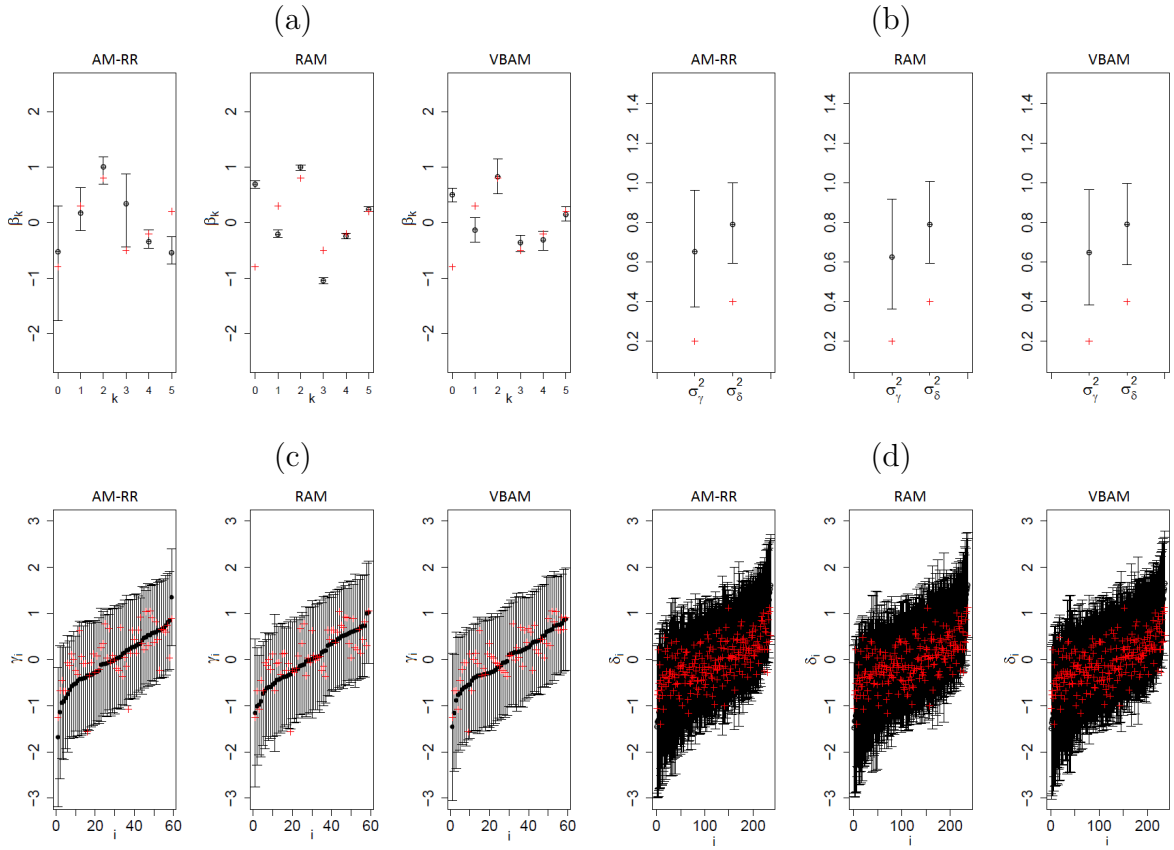


Figura D.2: *Cenário utilizando o ARS na estimação dos efeitos aleatórios* $\gamma_i$ *e* $\delta_{iv}$*. Valores reais ("+" vermelho), média a posteriori (círculo) e o intervalo HPD de 95% (barra) para os parâmetros do modelo Poisson log-linear. Os algoritmos AM-RR, RAM e VBAM (nessa ordem) são comparados em cada painel. Os intervalos nos Painéis (c) e (d) estão ordenados em ordem crescente com relação aos valores reais de* $\boldsymbol{\gamma}$ *e* $\boldsymbol{\delta}$*.*

# Apêndice E: Distribuições condicionais completas para o modelo Poisson log-linear com mistura

Apresentamos aqui as distribuições condicionais completas relacionadas ao modelo Poisson log-linear com mistura de duas distribuições normais para os efeitos aleatórios.

$$\begin{aligned}P(\gamma_i|\boldsymbol{\beta},\boldsymbol{\delta},\sigma^2_{\gamma_0},\sigma^2_{\gamma_1},W_i,\mathbf{y},\mathbf{X}) &\propto P(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta})\times P(\gamma_i)\\ &\propto \exp\{\sum_{v=1}^{M} Y_{iv}(\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv})\}\times\\ &\times \exp\{-\sum_{v=1}^{M} e^{\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv}} - \frac{1}{2\sigma^2_{\gamma_{W_i}}}\gamma_i^2\}.\end{aligned}$$

$$\begin{aligned}P(\delta_{iv}|\boldsymbol{\beta},\boldsymbol{\gamma},\sigma^2_{\delta_0},\sigma^2_{\delta_1},Q_{iv},\mathbf{y},\mathbf{X}) &\propto P(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta})\times P(\delta_{iv})\\ &\propto \exp\{Y_{iv}(\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv})\}\times\\ &\times \exp\{-e^{\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv}} - \frac{1}{2\sigma^2_{\delta_{Q_{iv}}}}\delta_{iv}^2\}.\end{aligned}$$

$$\begin{aligned}P(W_i|\boldsymbol{\beta},\boldsymbol{\delta},\boldsymbol{\gamma},\sigma^2_{\gamma_0},\sigma^2_{\gamma_1},\rho_{\gamma_1},\mathbf{y},\mathbf{X}) &\propto P(\gamma_i|\boldsymbol{\beta},\boldsymbol{\delta},\sigma^2_{\gamma_0},\sigma^2_{\gamma_1},W_i,\rho_{\gamma_1},\mathbf{y},\mathbf{X})\times P(W_i)\\ &\propto \exp\{\sum_{v=1}^{M} Y_{iv}(\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv})\}\times\\ &\times \exp\{-\sum_{v=1}^{M} e^{\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv}} - \frac{1}{2\sigma^2_{\gamma_{W_i}}}\gamma_i^2\times\\ &\times \rho_{\gamma_0}^{W_i}(1-\rho_{\gamma_0})^{1-W_i}. \qquad (7.1)\end{aligned}$$

Note que $(W_i|\boldsymbol{\beta},\boldsymbol{\delta},\boldsymbol{\gamma},\sigma^2_{\gamma_0},\sigma^2_{\gamma_1},\rho_{\gamma_1},\mathbf{y},\mathbf{X}) \sim \text{Bernoulli}(p^*_{W_i})$, com $p^*_{W_i} = \frac{P(W_i=1|-)}{P(W_i=0|-)+P(W_i=1|-)}$, sendo $P(W_i|-)$ a distribuição condicional completa de $W_i$ apresentada em 7.1.

$$\begin{aligned}P(Q_{iv}|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\sigma^2_{\delta_0},\sigma^2_{\delta_1},\rho_q,\mathbf{y},\mathbf{X}) &\propto P(\delta_{iv}|\boldsymbol{\beta},\boldsymbol{\gamma},\sigma^2_{\delta_0},\sigma^2_{\delta_1},Q_{iv},\rho_q,\mathbf{y},\mathbf{X})\times P(Q_{iv})\\ &\propto \exp\{Y_{iv}(\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv})\}\times\\ &\times \exp\{-e^{\mathbf{x}_{iv}\boldsymbol{\beta}+\gamma_i+\delta_{iv}} - \frac{1}{2\sigma^2_{\delta_{Q_{iv}}}}\delta_{iv}^2\}\times\\ &\times \rho_{\delta_0}^{Q_{iv}}(1-\rho_{\delta_0})^{1-Q_{iv}}.\end{aligned}$$

Note que $(Q_{iv}|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\sigma^2_{\delta_0},\sigma^2_{\delta_1},\rho_q,\mathbf{y},\mathbf{X}) \sim \text{Bernoulli}(p^*_{Q_{iv}})$, com $p^*_{Q_{iv}} = \frac{P(Q_{iv}=1|-)}{P(Q_{iv}=0|-)+P(Q_{iv}=1|-)}$.

$$(\rho_{\gamma_0}|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},W_i,\mathbf{y},\mathbf{X}) \sim \text{Beta}\left[a_1+\sum_{i=1}^{N} W_i, b_1+N-\sum_{i=1}^{N} W_i\right].$$

$$(\rho_{\delta_0}|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},Q_{iv},\mathbf{y},\mathbf{X}) \sim \text{Beta}\left[c_1+\sum_{i=1}^{N}\sum_{v=1}^{M} Q_{iv}, d_1+NM-\sum_{i=1}^{N}\sum_{v=1}^{M} Q_{iv}\right].$$

$$\left(\sigma^2_{\gamma_0}|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\sigma^2_{\gamma_1},W_i=0,\mathbf{y},\mathbf{X}\right) \sim GI\left[\frac{N_0^*}{2}+g_{1_0}, \frac{1}{2}(\sum_{i=1}^{N}(1-W_i)\gamma_i^2+g_{2_0})\right].$$

$$\left(\sigma^2_{\gamma_1}|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\sigma^2_{\gamma_0},W_i=1,\mathbf{y},\mathbf{X}\right) \sim GI\left[\frac{N_1^*}{2}+g_{1_1}, \frac{1}{2}(\sum_{i=1}^{N}W_i\gamma_i^2+g_{2_1})\right].$$

$$\left(\sigma^2_{\delta_0}|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\sigma^2_{\delta_1},Q_{iv}=0,\mathbf{y},\mathbf{X}\right) \sim GI\left[\frac{M_0^*}{2}+d_{1_0}, \frac{1}{2}(\sum_{i=1}^{N}\sum_{v=1}^{M}(1-Q_{iv})\delta_{iv}^2+d_{2_0})\right].$$

$$\left(\sigma^2_{\delta_1}|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\sigma^2_{\delta_0},Q_{iv}=1,\mathbf{y},\mathbf{X}\right) \sim GI\left[\frac{M_1^*}{2}+d_{1_1}, \frac{1}{2}(\sum_{i=1}^{N}\sum_{v=1}^{M}Q_{iv}\delta_{iv}^2+d_{2_1})\right].$$

com $N_0^* = \sum_{i=1}^{N}(1-W_i)$, $N_1^* = \sum_{i=1}^{N} W_i$, $M_0^* = \sum_{i=1}^{N}\sum_{v=1}^{M}(1-Q_{iv})$ e $M_1^* = \sum_{i=1}^{N}\sum_{v=1}^{M} Q_{iv}$.

# Referências Bibliográficas


Andrews, D. e Mallows, C. L. (1974), "Scale mixtures of Normal distributions," *Journal of the Royal Statistical Society B*, 36, 99–102.

Andrieu, C. e Thoms, J. (2008), "A tutorial on adaptive MCMC," *Statistics and Computing*, 18, 343–373.

Atchadé, Y. e Fort, G. (2010), "Limit theorems for some adaptive MCMC algorithms with subgeometric kernels," *International Statistical Institute (ISI) and Bernoulli Society for Mathematical Statistics and Probability*, 16, 116–154.

Booth, J., Casella, G., Friedl, H., e Hobert, J. (2003), "Negative binomial loglinear mixed models," *Statistical Modelling*, 3, 179–191.

Chen, M., Shao, Q., e Ibrahim, J. (2000), *Monte Carlo Methods in Bayesian Computation*, Springer-Verlag, New York.

Eilers, P. e Marx, B. (1996), "Flexible smoothing with B-splines and penalties (with Discussion)," *Statistical Science*, 11, 89–121.

Fang, K., Kotz, S., e Ng, K. (1990), *Symmetric multivariate and related distributions*, Chapman and Hall, London.

Feller, W. (1967), *An Introduction to Probability Theory and Its applications*, vol. 1, John Wiley, 3 edn.

Fotouhi, A. R. (2008), "Modelling overdispersion in longitudinal count data in clinical trials with an application to epileptic data," *Contemporary Clinical Trials*, 29, 547–554.

Gamerman, D. (1997), "Sampling from the posterior distribution in generalized linear mixed models," *Statistics and Computing*, 7, 57–68.

Gamerman, D. e Lopes, H. F. (2006), *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, vol. 68, Chapman and Hall/CRC, London, 2 edn.

Gelfand, A. e Smith, A. (1990), "Sampling based approaches to calculating marginal densities," *Journal of American Statistical Association*, 85, 398–409.

Gelman, A., Roberts, G., e Gilks, W. (1996), "Efficient Metropolis jumping rules," *Bayesian Statistics V*, pp. 599–608.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., e Rubin, D. (2014), *Bayesian Data Analysis*, Chapman and Hall/CRC, 3 edn.

Geman, S. e Geman, D. (1984), "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions Pattern Analysis and Machine Intelligence*, 6, 721–741.

Geweke, J. (1992), "Evaluating the accuracy of sampling-based approaches to calculating posterior moments," in *Bayesian Statistics*, pp. 169–193, University Press.

Gilks, W. (1992), "Derivative-free adaptive rejection sampling for Gibbs sampling," *In Bernardo, J., Berger, J., Dawid, A., and Smith, A., editors. Bayesian Statistics 4*, pp. 641–649. Oxford University Press, Oxford.

Gilks, W. e Wild, P. (1992), "Adaptive rejection sampling for Gibbs sampling," *Applied Statistics*, 41, 337–348.

Gilks, W., Best, N., e Tan, K. (1995), "Adaptive rejection Metropolis sampling within Gibbs Sampling," *Journal of Applied Statistics Series C*, 44, 455–472.

Haario, H., Saksman, E., e Tamminen, J. (1999), "Adaptive proposal distribution for random walk Metropolis algorithm," *Journal of Computing and Statistics*, 14, 375–395.

Haario, H., Saksman, E., e Tamminen, J. (2001), "An adaptive Metropolis algorithm," *International Statistical Institute (ISI) and Bernoulli Society for Mathematical Statistics and Probability*, 7, 223–243.

Hastings, W. (1970), "Monte Carlo sampling using Markov chains and their applications," *Biometrika*, 57, 97–109.

Heidelberger, P. e Welch, P. (1983), "Simulation run length control in the presence of an initial transient," *Operations Research*, 31, 1009–1044.

Kleinman, K. e Ibrahim, J. (1998), "A semi-parametric Bayesian approach to generalized linear mixed models," *Statistics in Medicine*, 17, 2579–2596.

Komárek, A. e Laseffre, E. (2008), "Generalized linear mixed model with a penalized Gaussian mixture as a random effects distribution," *Computational Statistics and Data Analysis*, 52, 3441–3458.

Lange, K. e Sinsheimer, J. (1993), "Normal/Independent distributions and their applications in robust regression," *Journal of Computational and Graphical Statistics*, 2, 75–198.

Marin, J. e Robert, C. (2007), *Bayesian Core: A Practical Approach to Computacional Bayesian Statistics*, vol. 1, Springer, New York.

Martino, L., Read, J., e Luengo, D. (2015), "Improved adaptive rejection Metropolis sampling algorithms," *IEEE Transactions on Signal Processing*, 63, 3123–3138.

Mbalawata, I. S., Särkkä, S., Vihola, M., e Haario, H. (2015), "Adaptive Metropolis algorithm using variational Bayesian adaptive Kalman Filter," *Computational Statistics and Data Analysis*, 83, 101–115.

McCullagh, P. e Nelder, J. (1989), *Generalized Linear Models*, Chapman and Hall, New York, 2 edn.

Mengersen, K. e Tweedie, R. (1996), "Rates of convergence of the Hastings and Metropolis algorithms," *Annals of Applied Statistics*, 24, 101–121.

Metropolis, N., Rosenbluth, A., Teller, M., e Teller, E. (1953), "Equations of state calculations by fast computing machines," *Journal of Chemistry and Physics*, 21, 1087–1091.

Muller, P. (1991), "A generic approach to posterior integration and Gibbs sampling," Technical report, Purdue University, West Lafayette, Indiana.

Muller, P. (1993), "Alternatives to the Gibbs Sampling scheme." Technical report, Institute of Statistics and Decision Science, Duke University, Durham, North Carolina.

Neal, R. (2003), "Slice sampling," *Annals of Statistics*, 31, 705–767.

Norris, J. (1998), *Markov Chains*, Cambridge University Press, Cambridge.

Plummer, M., Best, N., Cowles, K., e Vines, K. (2006), "CODA: convergence diagnosis and output analysis for MCMC," *R News*, 6, 7–11.

R Core Team (2015), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria.

Robert, C. (2007), *The Bayesian Choice from Decision-Theoretic Foundations to Computacional Implementation*, Springer, New York, 2 edn.

Robert, C. e Casella, G. (2004), *Monte Carlo Statistical Methods*, Springer, New York, 2 edn.

Robert, C. e Casella, G. (2009), *Introducing Monte Carlo Methods with R*, Springer, New York.

Roberts, G. e Rosenthal, R. (2009), "Examples of adaptive MCMC," *Journal of Computational and Graphical Statistics*, 18, 349–367.

Roberts, G., Gelman, A., e Gilks, W. (1997), "Weak convergence and optimal scaling of random walk Metropolis algorithm," *Annals of Applied Statistics*, 7, 110–120.

Serfling, R. (2006), "Multivariate symmetry and asymmetry," *Encyclopedia of Statistical Sciences, Second Edition (S. Kotz, N. Balakrishnan, C. B. Read and B. Vidakovic, eds.)*, 8, 5338–5345. Wiley.

Spiegelhalter, D., Best, N., Carlin, B., e van der Linde, A. (2002), "Bayesian measures of model complexity and fit (with discussion)," *Journal of the Royal Statistical Society B*, 64, 583–639.

Särkkä, S. e Hartikainen, J. (2013), "Non-linear noise adaptive Kalman filtering via variational Bayes," *Proceedings of Machine Learning for Signal Processing*, pp. 1–6.

Särkkä, S. e Nummenmaa, A. (2009), "Recursive noise adaptive Kalman filtering by variational Bayesian approximations," *IEEE Transactions on Automatic Control*, 54, 596–600.

Thall, P. e Vail, S. (1990), "Some covariance models for longitudinal count data with overdispersion," *Biometrics*, 18, 657–671.

Vihola, M. (2012), "Robust adaptive Metropolis algorithm with coerced acceptance rate," *Statistics and Computing*, 22, 997–1008.

Watanabe, S. (2002), "Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory," *Journal of Machine Learning Research*, 11, 3571–3594.

West, M. (1985), "Generalized linear models: outlier accomodation, scale parameters and prior distribution," *Bayesian Statistics 2*, pp. 531 – 558.