

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS

**Comparação entre algoritmos MCMC para Inferência Bayesiana em  
modelos dicotômicos da TRI**

Orientador: Flávio Bambirra Gonçalves

Gabriel Oliveira Assunção

Belo Horizonte

2019

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS

**Comparação entre algoritmos MCMC para Inferência Bayesiana em  
modelos dicotômicos da TRI**

Orientador: Flávio Bambirra Gonçalves

Gabriel Oliveira Assunção

Belo Horizonte

Junho 2019



# Resumo

Este estudo tem o objetivo de comparar algoritmos de MCMC para Inferência Bayesiana no modelo da TRI de 3 parâmetros. São considerados quatro diferentes algoritmos já propostos na literatura, que se diferenciam basicamente em relação ao uso de variáveis auxiliares. O objetivo principal é investigar qual algoritmo é computacionalmente mais eficiente para retornar uma amostra da (mesma) distribuição *a posteriori*. A comparação é feita com base em tempo computacional e tamanho efetivo de amostra de estatísticas relevantes. A comparação é feita em diferentes cenários com relação à tamanho de amostra (incluindo os itens). Através dela, pode-se perceber que a performance dos algoritmos varia conforme o tamanho de amostra aumenta. E apresentado também uma extensão do algoritmo de Gonçalves et al. (2018) para o modelo de 4 parâmetros e este é aplicado a um banco de dados do Enem.

# Abstract

This study aims to compare MCMC algorithms for Bayesian Inference in the 3-parameter TRI model. We consider four different algorithms already proposed in the literature, which differ basically in relation to the use of auxiliary variables. The main objective is to investigate which algorithm is computationally more efficient to return a sample of the (same) distribution to textit posteriori. The comparison is made based on computational time and effective sample size of relevant statistics. The comparison is made in different scenarios with respect to sample size (including items). Through it, one can see that the performance of the algorithms varies as the sample size increases. An extension of the Gonçalves et al. (2018) algorithm for the 4-parameter model is also presented and applied to an Enem database.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Teoria de Resposta ao Item . . . . .	2
1.2	Inferência Bayesiana . . . . .	6
1.3	Markov Chain Monte Carlo . . . . .	7
1.3.1	Gibbs Sampling . . . . .	7
1.3.2	Metropolis-Hastings . . . . .	7
<b>2</b>	<b>Metodologia</b>	<b>9</b>
2.1	Algoritmo Béguin and Glas (2001) . . . . .	9
2.2	Algoritmo Gonçalves et al. (2018) . . . . .	12
2.3	Algoritmo Metropolis-Hastings I . . . . .	14
2.4	Algoritmo Metropolis-Hastings II . . . . .	17
2.5	Critérios para comparação . . . . .	19
<b>3</b>	<b>Simulações</b>	<b>22</b>
3.1	Estudo simulado 1 . . . . .	23
3.2	Estudo simulado 2 . . . . .	27
3.3	Estudo simulado 3 . . . . .	30
3.4	Comparação entre simulações . . . . .	33
<b>4</b>	<b>Extensão do algoritmo de Gonçalves et al. (2018) para modelo 4 parâmetros</b>	<b>39</b>
4.1	Algoritmo . . . . .	39
4.2	Estudo simulado . . . . .	42
4.2.1	Estudo simulado 1 . . . . .	42
4.2.2	Estudo simulado 2 . . . . .	44
4.3	Comparação entre os modelos de 3 e 4 parâmetros . . . . .	47
<b>5</b>	<b>Aplicação dos algoritmos ao banco de dados do Enem</b>	<b>53</b>
<b>6</b>	<b>Conclusão</b>	<b>58</b>

<b>A</b>	<b>Cálculos</b>	<b>62</b>
A.1	Cálculos Algoritmo Béguin and Glas (2001) . . . . .	62
A.2	Cálculos Algoritmo Gonçalves et al. (2018) . . . . .	65
A.3	Cálculos Algoritmo Metropolis-Hastings I . . . . .	67
A.4	Cálculos Algoritmo Metropolis-Hastings II . . . . .	69
<b>B</b>	<b>Cálculos Algoritmo Modelo de 4 parâmetros</b>	<b>69</b>
<b>C</b>	<b>Cadeias de Markov</b>	<b>71</b>
C.1	Estudo Simulado 1 . . . . .	71
C.2	Estudo Simulado 2 . . . . .	73
C.3	Estudo Simulado 3 . . . . .	75

## Lista de Tabelas

1	Número de itens e indivíduos por simulação . . . . .	22
2	Valores iniciais de cada parâmetro em cada repetição . . . . .	23
3	Tempo computacional, médio, por iteração e até se obter o burn-in por algoritmo para estudo simulado 1. . . . .	24
4	Tamanho Efetivo de Amostra após 10000 iterações para cada repetição no Estudo Simulado 1. . . . .	25
5	Razão entre os tempos médios por iteração de cada algoritmo para Estudo Simulado 1. . . . .	25
6	Razão entre o tamanho efetivo de amostra médio de cada algoritmo para Estudo Simulado 1. . . . .	25
7	Número de iterações e tempo computacional necessário para se obter um tamanho efetivo de amostra fixo para Estudo Simulado 1. . . . .	26
8	Razão entre os tempos por iteração das repetições de melhor desempenho de cada algoritmo para Estudo Simulado 1. . . . .	26
9	Razão entre o tamanho efetivo de amostra das repetições de melhor desempenho de cada algoritmo para Estudo Simulado 1. . . . .	27
10	Número de iterações e tamanho efetivo de amostra, da repetição de melhor desempenho, com tempo computacional fixo para Estudo Simulado 1. . . . .	27
11	Tempo computacional, médio, por iteração e até se obter o burn-in por algoritmo para estudo simulado 2 . . . . .	27
12	Tamanho Efetivo de Amostra após 10000 iterações para cada repetição no Estudo Simulado 2. . . . .	28
13	Razão entre os tempos médios por iteração de cada algoritmo para Estudo Simulado 2. . . . .	29
14	Razão entre o tamanho efetivo de amostra médio de cada algoritmo para Estudo Simulado 2. . . . .	29
15	Número de iterações e tempo computacional necessário para se obter um tamanho efetivo de amostra fixo para Estudo Simulado 2. . . . .	29



16	Razão entre os tempos por iteração das repetições de melhor desempenho de cada algoritmo para Estudo Simulado 2. . . . .	30
17	Razão entre o tamanho efetivo de amostra das repetições de melhor desempenho de cada algoritmo para Estudo Simulado 2. . . . .	30
18	Número de iterações e tamanho efetivo de amostra, da repetição de melhor desempenho, com tempo computacional fixo para Estudo Simulado 2. . . . .	30
19	Tempo computacional, médio, por iteração e até se obter o burn-in por algoritmo para estudo simulado 3 . . . . .	30
20	Tamanho Efetivo de Amostra após 10000 iterações para cada repetição no Estudo Simulado 3. . . . .	31
21	Razão entre os tempos médios por iteração de cada algoritmo para Estudo Simulado 3. . . . .	32
22	Razão entre o tamanho efetivo de amostra médio de cada algoritmo para Estudo Simulado 3. . . . .	32
23	Número de iterações e tempo computacional necessário para se obter um tamanho efetivo de amostra fixo para Estudo Simulado 3 . . . . .	33
24	Razão entre os tempos por iteração das repetições de melhor desempenho de cada algoritmo para Estudo Simulado 3. . . . .	33
25	Razão entre o tamanho efetivo de amostra das repetições de melhor desempenho de cada algoritmo para Estudo Simulado 3. . . . .	33
26	Número de iterações e tamanho efetivo de amostra, da repetição de melhor desempenho, com tempo computacional fixo para Estudo Simulado 3. . . . .	33
27	Maiores diferenças de colocação . . . . .	55
28	Diferença entre indivíduos top 100 . . . . .	57

## Lista de Figuras

1	Curva Característica do Item . . . . .	3
2	Influência do parâmetro de discriminação . . . . .	4
3	Influência do parâmetro de dificuldade . . . . .	4
4	Influência do parâmetro de acerto casual . . . . .	5
5	Influência do parâmetro assíntota superior . . . . .	5
6	Comparação da função de autocorrelação da log-densidade a posteriori entre os quatro algoritmos no Estudo Simulado 1. . . . .	24
7	Comparação da função de autocorrelação da log-densidade a posteriori entre os quatro algoritmos no Estudo Simulado 2. . . . .	28
8	Comparação da função de autocorrelação da log-densidade a posteriori entre os quatro algoritmos no Estudo Simulado 3. . . . .	31
9	Comparação tempo por iteração das melhores performances de cada algoritmo. . . . .	34
10	Comparação razão entre o tempo por iteração de cada algoritmo e o máximo de tempo por iteração observado em cada estudo. . . . .	35
11	Comparação tamanho efetivo de amostra das melhores performances de cada algoritmo. . . . .	35
12	Comparação do tempo, em segundos, para a melhores performances de cada algoritmo obter um tamanho efetivo de amostra fixo. . . . .	36
13	Comparação razão entre o tempo para se obter um tamanho efetivo de amostra fixo de cada algoritmo e o máximo de tempo para se obter um tamanho efetivo de amostra fixo de cada estudo. . . . .	37
14	Comparação do tamanho efetivo da amostra, das melhores performances de cada algoritmo, considerando tempo fixo. . . . .	38
15	Comparação do número de iterações, das melhores performances de cada algoritmo, considerando tempo fixo. . . . .	38
16	Estimativas dos parâmetros na primeiro estudo simulado. . . . .	43
17	Distribuição do valor real e estimativa dos traços latentes para primeiro estudo simulado. . . . .	44

18	CCI item 29 Estudo Simulado 1 . . . . .	44
19	Estimativas dos parâmetros para segundo estudo simulado. . . . .	45
20	Distribuição do valor real e estimativa dos traços latentes para segundo estudo simulado. . . . .	46
21	CCI item 50 Estudo Simulado 2 . . . . .	46
22	Estimativas para parâmetros dos itens ajustando o modelo 4PNO aos dados gerados pelo 3PNO. . . . .	47
23	Estimativas para o traço latente ajustando o modelo 4PNO aos dados gerados pelo 3PNO. . . . .	48
24	Distribuição das estimativas para o traço latente ajustando o modelo 4PNO aos dados gerados pelo modelo 3PNO. . . . .	48
25	Estimativas para assintota superior ajustando o modelo 4PNO aos dados gerados pelo 3PNO. . . . .	49
26	CCI para 41° item . . . . .	49
27	Estimativas para parâmetros dos itens ajustando o modelo 3PNO aos dados gerados pelo 4PNO. . . . .	50
28	Estimativas para o traço latente ajustando o modelo 3PNO aos dados gerados pelo 4PNO. . . . .	51
29	Distribuição das estimativas para o traço latente ajustando o modelo 3PNO aos dados gerados pelo modelo 4PNO. . . . .	51
30	CCI para 13° item . . . . .	52
31	CCI para 30° item . . . . .	52
32	Estimativas para os parâmetros $a$ , $b$ , $c$ e $\theta$ utilizando os modelos de 3 parâmetros e 4 parâmetros ao banco de dados Enem. . . . .	54
33	Distribuição das estimativas para o traço latente ajustando o modelo 3PNO e modelo 4PNO. . . . .	54
34	Estimativa para o parâmetro $d$ no banco de dados Enem. . . . .	55
35	CCI item 19 do banco de dados Enem . . . . .	56
36	Diferença da colocação dos candidatos em relação ao modelo de 3 parâmetros com o modelo de 4 parâmetros . . . . .	56

37	Cadeias Log <i>Posteriori</i> algoritmo Béguin and Glas (2001) para Estudo Simulado 1. . . . .	71
38	Cadeias Log <i>Posteriori</i> algoritmo Gonçalves et al. (2018) para Estudo Simulado 1. . . . .	72
39	Cadeias Log <i>Posteriori</i> algoritmo Metropolis I para Estudo Simulado 1. . . . .	72
40	Cadeias Log <i>Posteriori</i> algoritmo Metropolis II para Estudo Simulado 1. . . . .	73
41	Cadeias Log <i>Posteriori</i> algoritmo Béguin and Glas (2001) para Estudo Simulado 2. . . . .	74
42	Cadeias Log <i>Posteriori</i> algoritmo Gonçalves et al. (2018) para Estudo Simulado 2. . . . .	74
43	Cadeias Log <i>Posteriori</i> algoritmo Metropolis I para Estudo Simulado 2. . . . .	75
44	Cadeias Log <i>Posteriori</i> algoritmo Metropolis II para Estudo Simulado 2. . . . .	75
45	Cadeias Log <i>Posteriori</i> algoritmo Béguin and Glas (2001) para Estudo Simulado 3. . . . .	76
46	Cadeias Log <i>Posteriori</i> algoritmo Gonçalves et al. (2018) para Estudo Simulado 3. . . . .	76
47	Cadeias Log <i>Posteriori</i> algoritmo Metropolis I para Estudo Simulado 3. . . . .	77
48	Cadeias Log <i>Posteriori</i> algoritmo Metropolis II para Estudo Simulado 3. . . . .	77

# 1 Introdução

A Teoria da Resposta ao Item (TRI) teve suas ideias fundamentadas na área da psicomетria por Binet and Simon (1904), logo após a publicação de Spearman (1904), artigo que se tornou a base para a Teoria Clássica dos Testes. A diferença básica entre os dois é que, enquanto Binet and Simon considera a existência de uma variável latente para se estimar a "habilidade" de um indivíduo, Spearman considera apenas que a sua estimativa se dá através do score real e um erro de medição. Apenas na década de 20, o trabalho de Binet and Simon foi reconhecido, sendo exploradas suas ideias no trabalho de Thurstone (1925).

Apenas na década de 50, surgiram os primeiros modelos probabilísticos para o princípio da TRI, onde sua formulação era baseada na probabilidade de um indivíduo acertar um item condicionado na sua habilidade, medida não observável. Em 1952, Lord propõe sua construção de modelo utilizando a função acumulada da distribuição Normal (ogiva Normal) conhecida como modelo de 2 parâmetros ogiva normal (2PNO). Em 1960, Rash propõe um modelo utilizando a função logística conhecida com 1 parâmetro logístico (1PL).

Birnbaum (1968) contribui para o trabalho de Lord, ao propor a inclusão de um parâmetro de acerto casual para o modelo, criando assim, o conhecido modelo de 3 parâmetros logístico (3PL), este modelo atribui uma probabilidade de acertar o item para um indivíduo com baixa habilidade, caracterizando uma assíntota inferior. Barton and Lord (1981) propõem um modelo de 4 parâmetros (4PL), que inclui uma assíntota superior à curva do modelo, para que assim um aluno com habilidade alta tenha uma certa probabilidade de errar. Neste novo modelo os indivíduos com habilidade alta não seriam tão penalizados ao errar um item fácil quanto no 3PL.

É de interesse estimar os parâmetros dos itens e as habilidades dos indivíduos, para isso, o pesquisador pode seguir duas vertentes, a Inferência Clássica e a Inferência Bayesiana. Na Inferência Clássica, são geralmente utilizados algoritmos EM. Para a Bayesiana, são considerados algoritmos de Markov Chain Monte Carlo (MCMC).

Os algoritmos de MCMC mais utilizados para Inferência Bayesiana surgiram nas décadas de 50 e 80, o primeiro destes algoritmos, o Metropolis-Hastings teve seu início em um algoritmo proposto por Metropolis et al. (1953) e foi generalizado por Hastings (1970). O

segundo algoritmo é o algoritmo de Gibbs Sampling proposto por Geman and Geman (1984).

A construção de algoritmos MCMC para modelos dicotômicos de TRI podem variar em diferentes aspectos, como esquema de blocagem, passos Metropolis-Hastings, uso de variáveis auxiliares, etc. Dentre os algoritmos existentes na literatura para os modelos de TRI destacam-se aqueles que utilizam de variáveis auxiliares para evitar a utilização de um passo Metropolis-Hastings, em particular, para o modelo de 3PNO, os algoritmos de Béguin and Glas (2001) e Gonçalves et al. (2018), ambos extensão do algoritmo proposto por Albert (1992) para o modelo 2PNO. Para o modelo 4PNO, existe na literatura o algoritmo de Culpepper (2016), extensão do algoritmo de Béguin and Glas (2001).

O objetivo deste trabalho é comparar os algoritmos existentes na literatura para o modelo de 3 parâmetros no custo computacional de cada um, observando características como tempo computacional e tamanho efetivo de amostra. As comparações são feitas através de estudos simulados, em que os algoritmos considerados são aplicados à base de dados simulados para análise de sua performance.

Também será proposta uma extensão do algoritmo de Gonçalves et al. para o modelo de 4 parâmetros. São realizadas análises em dados simulados e em uma base de dados do Exame Nacional do Ensino Médio (ENEM).

## 1.1 Teoria de Resposta ao Item

Os modelos utilizados neste trabalho são o 3PNO e 4PNO, sendo este último uma generalização do modelo anterior. No modelo 4 PNO são atribuídos quatro parâmetros para o item, sendo estes parâmetros discriminação, dificuldade, acerto casual (assíntota inferior) e assíntota superior, para o modelo 3PNO são atribuídos apenas os três primeiros parâmetros para os itens. Uma das premissas da TRI é a suposição de que os itens são independentes entre si e que os indivíduos são independentes entre si, quando condicionados aos parâmetros dos itens e as habilidades dos indivíduos. O modelo 4PNO modela a probabilidade de um indivíduo  $j$  acertar o item  $i$ , da seguinte forma:

$$P(Y_{ij} = 1|a_i, b_i, c_i, d_i, \theta_j) = c_i + (1 - d_i - c_i)\Phi(a_i(\theta_j - b_i)), \quad (1)$$

onde,  $a_i$  é a discriminação do  $i$ -ésimo item,  $b_i$  é a dificuldade do  $i$ -ésimo item,  $c_i$  é o acerto casual do  $i$ -ésimo item,  $d_i$  é a assíntota superior (para o caso 3PNO, a este parâmetro é atribuído valor 0),  $\theta_j$  é o traço latente (habilidade) do  $j$ -ésimo indivíduo e  $\Phi(\cdot)$  representa a função acumulada da distribuição Normal padrão. Uma outra alternativa ao 4PNO é o 4PL, no qual ao invés de se utilizar  $\Phi(a_i(\theta_j - b_i))$  utiliza-se  $e^{-Da_i(\theta_j - b_i)}$ . Quando  $D=1.7$ , ambos os modelos têm valores de probabilidade estritamente próximos.

Uma maneira de representar esta probabilidade, facilitando a visualização e a interpretação, é através da Curva Característica do Item (CCI), Figura 1, onde fica representada a probabilidade de um indivíduo acertar o item (eixo Y) para diferentes traços latentes (eixo X).

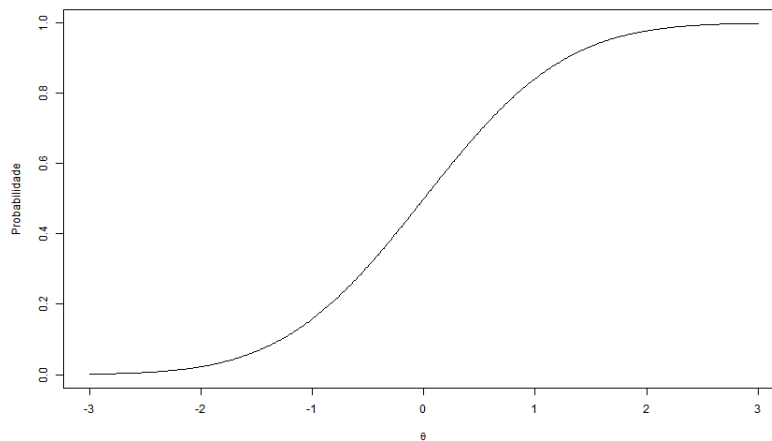


Figura 1: Curva Característica do Item

O parâmetro de discriminação influencia na inclinação da curva em uma determinada faixa de habilidade, quanto maior o seu valor, mais o item discrimina os indivíduos. Quando este parâmetro assume valor negativo, a CCI é decrescente. Na Figura 2, está representado como este parâmetro influencia na CCI.

Um problema inerente ao modelo (1) é a sua falta de identificabilidade. A maneira mais usual de torná-lo identificável, e a que será adotada, é a atribuição de uma distribuição Normal com parâmetros conhecidos, que basicamente fixa a escala, para os traços latentes. O parâmetro de dificuldade está na mesma escala que o traço latente, logo  $b_i \in R$ . Este parâmetro pode ser interpretado como um parâmetro de locação para a curva. Quando

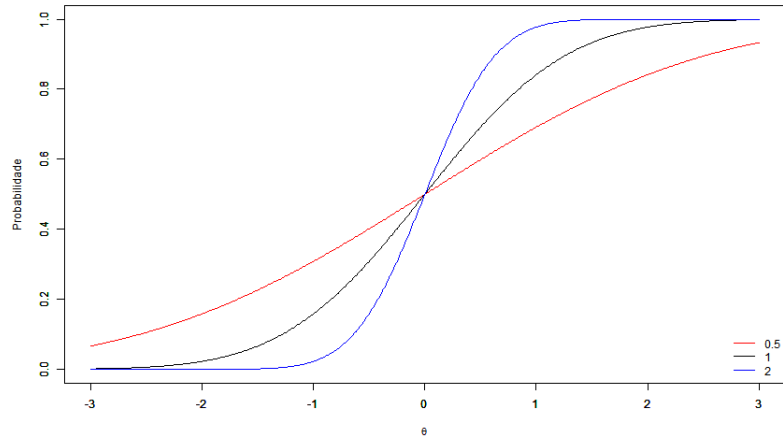


Figura 2: Influência do parâmetro de discriminação

$\theta_j = b_i$ ,  $P(Y_{ij} = 1 | a_i, c_i, d_i, b_i) = \frac{1 - d_i + c_i}{2}$ . Se  $b_i > \theta_j$  então o item é dito mais difícil de acertar, o oposto é dito que o item é mais fácil. Na Figura 3, está representado o impacto deste parâmetro na CCI.

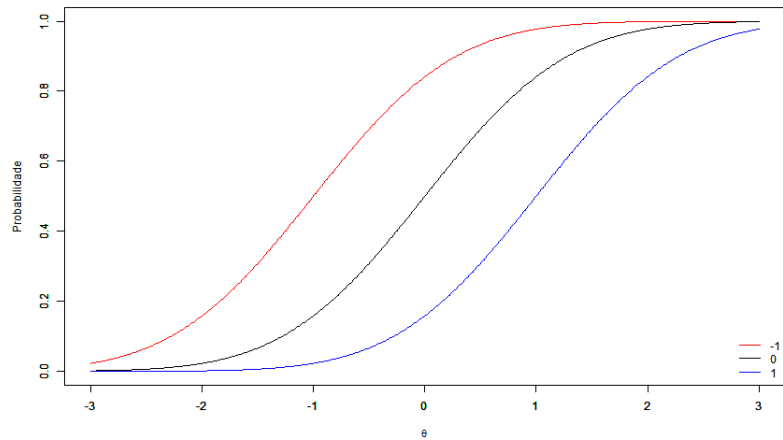


Figura 3: Influência do parâmetro de dificuldade

O parâmetro de acerto casual é interpretado como a probabilidade de um indivíduo com habilidade baixa acertar o item, como é uma probabilidade temos que  $c_i \in [0, 1]$ . Este parâmetro é a assíntota inferior da CCI. A Figura 4 representa a curva CCI para diferentes assíntotas inferiores.

O parâmetro de assíntota superior representa a probabilidade de um indivíduo com



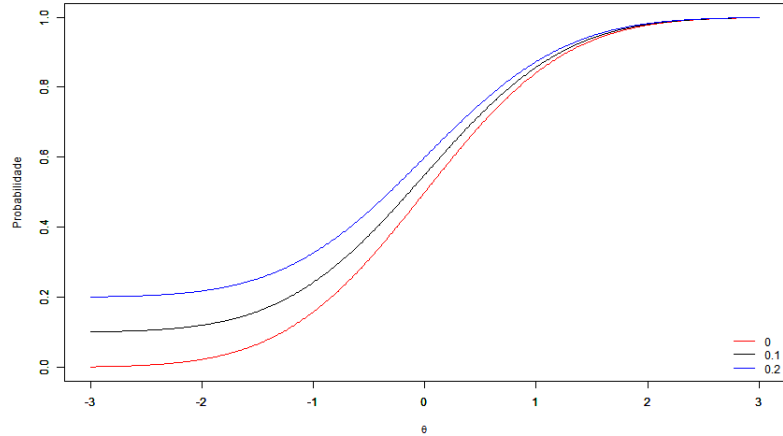


Figura 4: Influência do parâmetro de acerto casual

habilidade alta errar o item, como é uma probabilidade temos que  $d_i \in [0, 1]$ . Os parâmetros de acerto casual e assíntota superior devem satisfazer a seguinte condição  $c_i + d_i < 1$ . A Figura 5 representa diferentes curvas CCI para diferentes assíntotas superiores.

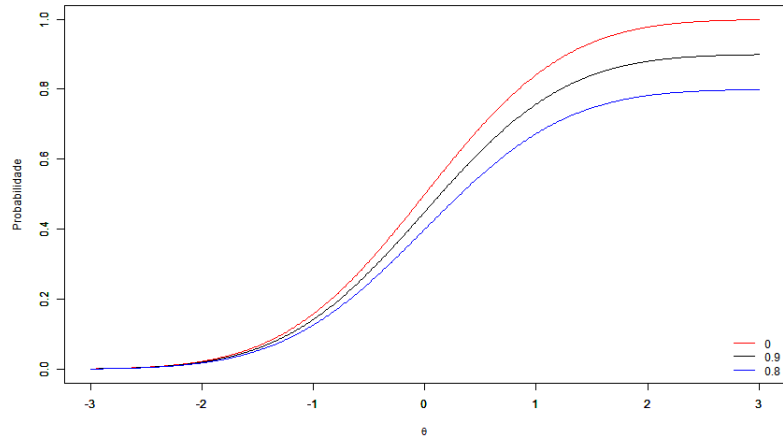


Figura 5: Influência do parâmetro assíntota superior

Para este trabalho será considerada a seguinte reparametrização de equação (1) para a construção dos algoritmos.

$$P(Y_{ij} = 1|a_i, b_i, c_i, d_i, \theta_j) = c_i + (1 - d_i - c_i)\Phi(a_i\theta_j - b_i). \quad (2)$$

## 1.2 Inferência Bayesiana

A inferência Bayesiana é fundamentada na interpretação subjetiva de probabilidade, na qual esta é utilizada para se mensurar incerteza. O Paradigma Bayesiano diz que toda inferência deve ser baseada na distribuição a *posteriori* de todas as quantidades desconhecidas do modelo condicionadas nos dados.

A inferência se dá na junção de duas fontes de informação, sendo elas os dados e a distribuição atribuída às quantidades desconhecidas (inclusive parâmetros) do modelo, esta é conhecida como distribuição a *priori*. Para um conjunto de variáveis  $Y$  a serem observadas, atribui-se uma distribuição com Função Densidade de Probabilidade ou Função Massa de Probabilidade  $\pi(Y|\theta)$ , condicionada a um conjunto de parâmetros desconhecidos  $\theta$ . Para uma distribuição à *priori* com FDP/FMP  $\pi(\theta)$  e um conjunto de dados  $Y = y$ , a distribuição a *posteriori* de  $\theta$  é definida como sendo a distribuição condicional de  $\theta$  dado  $Y = y$  e tem FDP/FMP dado por:

$$\pi(\theta|y) = \frac{\pi(y|\theta)\pi(\theta)}{\pi(y)}, \quad (3)$$

onde  $\pi(y) = \int_{-\infty}^{\infty} \pi(y|\theta)\pi(\theta)d\theta$ , é conhecida como distribuição preditiva à *priori*.

Uma vez que se tem a distribuição a *posteriori*, é desejado obter uma estimativa do parâmetro, podendo esta ser pontual ou intervalar. Para estimativas pontuais são comumente utilizadas a esperança, moda ou mediana, e para intervalar são utilizados intervalos de credibilidade que representam a probabilidade do parâmetro estar neste intervalo.

Nem sempre essas estimativas são fáceis de serem obtidas, podendo ter problema com a dimensão dos dados e de  $\theta$  ou quando a distribuição a *posteriori* não é tratável analiticamente. Para estes casos são utilizados algoritmos para se obter uma amostra da distribuição a *posteriori* e assim estimar esperanças com respeito a esta distribuição via Monte Carlo. Os algoritmos mais popularmente utilizados devido à sua flexibilidade e eficiência são os algoritmos MCMC. Esses se baseiam na construção de cadeias de Markov que têm a *posteriori* alvo como distribuição limite. Os algoritmos mais populares são o Gibbs Samplings e Metropolis-Hastings, sendo estes bastante flexíveis.

## 1.3 Markov Chain Monte Carlo

Algoritmos de MCMC são utilizados para gerar uma amostra (aproximada) grande de uma certa distribuição, quando não se consegue gerar diretamente desta distribuição. Toda sua construção se dá através de uma cadeia de Markov, com uma série de propriedades que precisam ser satisfeitas para que o algoritmo tenha as propriedades desejadas. Abaixo será apresentada a estrutura dos dois algoritmos mais utilizados na abordagem Bayesiana.

### 1.3.1 Gibbs Sampling

O algoritmo de Gibbs Sampling é baseado nas distribuições condicionais completas de suas componentes. Querendo se amostrar de  $\psi = (\psi_1, \psi_2, \dots, \psi_n)$  um conjunto de  $n$  variáveis aleatórias, de sua distribuição conjunta, o algoritmo funciona da seguinte maneira:

1. É atribuído valor inicial para as variáveis,  $\psi^0$ .
2. Para  $i$  de 1 até o número de iterações desejadas, gera valores de  $\psi_j^i$  seguindo a distribuição  $f$ :

$$\psi_1^i \sim f(\psi_1^i | \psi_2^{i-1}, \psi_3^{i-1}, \dots, \psi_n^{i-1}, y)$$

$$\psi_2^i \sim f(\psi_2^i | \psi_1^i, \psi_3^{i-1}, \dots, \psi_n^{i-1}, y)$$

.

.

.

$$\psi_n^i \sim f(\psi_n^i | \psi_1^i, \psi_2^i, \dots, \psi_{n-1}^i, y).$$

Na abordagem Bayesiana tem-se que  $\psi$  é o conjunto de parâmetros e variáveis latentes desejados a amostrar da distribuição *posteriori*, é necessário saber a distribuição condicional completa a *posteriori* de cada parâmetro isto é  $f(\psi_j | \psi_1, \psi_2, \dots, \psi_{j-1}, \psi_{j+1}, \dots, \psi_n, y)$ , onde  $y$  são os conjuntos de dados.

### 1.3.2 Metropolis-Hastings

Quando não se é possível amostrar diretamente de alguma condicional completa ( $f$ ), é utilizado o algoritmo Metropolis-Hastings, onde um valor é proposto de uma densidade

$h$ , que pode depender do estado atual da cadeia, e aceito com uma probabilidade que preserve a condicional completa como distribuição invariante da cadeia. Denotaremos  $h(x, y)$  a densidade  $h$  no ponto  $y$  condicionado em  $x$ . O algoritmo funciona da seguinte maneira:

1. É dado um valor inicial para  $x^0$ .
2. Para  $i$  de 1 até o número de iterações faça:  
Propõe um valor  $y \sim h(x)$ , onde  $x$  é o valor atual da cadeia, e calcula-se a probabilidade de aceitação  $\alpha(x, y)$ .
3. Aceita  $y$  com probabilidade:

$$\alpha(x, y) = \min \left( 1, \frac{f(y) h(y, x)}{f(x) h(x, y)} \right) \quad (4)$$

e rejeita  $y$  com probabilidade  $1 - \alpha(x, y)$ .

Pode-se perceber que para este algoritmo não é necessário conhecer as constantes de normalização da função  $f$ .

Há diversas maneiras de se escolher a distribuição proposta  $h(\cdot)$ . A opção mais comum é o passeio aleatório (Random-Walk), onde  $h$  é uma função par, isto é,  $h(x, y) = h(y, x)$ , um exemplo é a distribuição Normal com média  $x$ .

No caso do passeio aleatório Normal, sua variância deve ser escolhida de forma a otimizar a velocidade de convergência das cadeias. Isto é feito através de calibração da taxa de aceitação da cadeia, que deve estar em torno de 0.44 (em casos unidimensionais) e 0.234 (em casos multidimensionais), ver Roberts et al. (1997).

## 2 Metodologia

Nesta seção serão apresentados todos os quatro algoritmos MCMC para o modelo 3PNO que serão comparados.

### 2.1 Algoritmo Béguin and Glas (2001)

No algoritmo proposto por Béguin and Glas (2001), baseado no trabalho de Albert (1992) para o modelo de 2 parâmetros, são utilizadas duas variáveis auxiliares para a construção de um algoritmo de Gibbs-Samplings, no qual é possível amostrar diretamente de todas as distribuições condicionais completas.

Temos que o modelo 3PNO pode ser reescrito da seguinte forma:

$$\begin{aligned} P(Y_{ij} = 1|a_i, b_i, c_i, \theta_j) &= c_i + (1 - c_i)\Phi(a_i\theta_j - b_i) \\ &= \Phi(a_i\theta_j - b_i) + c_i(1 - \Phi(a_i\theta_j - b_i)). \end{aligned} \tag{5}$$

Os autores consideram uma variável auxiliar  $Z_{ij} \sim \text{Bernoulli}(\Phi(a_i\theta_j - b_i))$  e reescreve o modelo em (5) como:

$$\begin{aligned} (Y_{ij}|Z_{ij} = 1, c_i) &\sim \text{Bernoulli}(1) \\ (Y_{ij}|Z_{ij} = 0, c_i) &\sim \text{Bernoulli}(c_i) \end{aligned}$$

Pode-se mostrar que esta representação preserva a distribuição marginal em (4).

$$\begin{aligned} P(Y_{ij} = 1|a_i, b_i, c_i, \theta_j) &= P(Y_{ij} = 1|c_i, Z_{ij} = 1)P(Z_{ij} = 1|a_i, b_i, \theta_j) \\ &\quad + P(Y_{ij} = 1|c_i, Z_{ij} = 0)P(Z_{ij} = 0|a_i, b_i, \theta_j) \\ &= 1\Phi(a_i\theta_j - b_i) + c_i(1 - \Phi(a_i\theta_j - b_i)). \end{aligned}$$

Desta forma, os autores podem utilizar a metodologia utilizada por Albert (1992) e introduzir as variáveis  $X_{ij} \sim N(a_i\theta_j - b_i, 1)$ , de forma que  $Z_{ij} = 1$  se  $X_{ij} > 0$  e  $Z_{ij} = 0$  se  $X_{ij} \leq 0$ .

Desta maneira, os parâmetros que se tem interesse de simular no algoritmo de Gibbs Sampling são  $\Psi = \{\mathbf{Z}, \mathbf{X}, \mathbf{a}, \mathbf{b}, \mathbf{c}, \theta\}$ , onde  $\mathbf{c} = (c_1, \dots, c_I)$ ,  $\mathbf{a} = (a_1, \dots, a_I)$ ,  $\mathbf{b} = (b_1, \dots, b_I)$ ,

$\theta = (\theta_1, \dots, \theta_J)$ ,  $\mathbf{X}, \mathbf{Z}$  são matrizes de tamanho  $I \times J$  com entradas  $X_{ij}$  e  $Z_{ij}$  com  $i=1, \dots, I$  e  $j=1, \dots, J$ .

Para o algoritmo de MCMC, que converge para a distribuição a *posteriori* de  $\Psi$ , é utilizada a seguinte blocagem para simulação:  $\Psi_1 = \{\mathbf{Z}, \mathbf{X}\}$ ,  $\Psi_2 = \{\theta\}$ ,  $\Psi_3 = \{\mathbf{a}, \mathbf{b}\}$ ,  $\Psi_4 = \{\mathbf{c}\}$ . Como citado na seção 1.3.1, o algoritmo Gibbs Samplings amostra diretamente da distribuição condicional completa de cada bloco. Denotaremos a condicional completa de \* por  $\pi(*|\cdot)$ . Assim, utilizando a independência condicional das respostas, tem-se que:

$$\pi(\mathbf{X}, \mathbf{Z}|\cdot) = \prod_{i=1}^I \prod_{j=1}^J \pi(X_{ij}, Z_{ij}|\cdot),$$

$$\pi(\mathbf{a}, \mathbf{b}|\cdot) = \prod_{i=1}^I \pi(a_i, b_i|\cdot),$$

$$\pi(\mathbf{c}|\cdot) = \prod_{i=1}^I \pi(c_i|\cdot),$$

$$\pi(\theta|\cdot) = \prod_{j=1}^J \pi(\theta_j|\cdot).$$

Para o bloco  $\Psi_1$ :

$$\pi(Z_{ij}|\cdot) \propto \pi(Y_{ij}|Z_{ij})\pi(Z_{ij}).$$

Através de cálculos, apresentados no Apêndice A.1, é possível perceber que  $(Z_{ij}|Y_{ij} = 1, \cdot) \sim \text{Bernoulli}\left(\frac{\Phi(a_i\theta_j - b_i)}{\Phi(a_i\theta_j - b_i) + c_i(1 - \Phi(a_i\theta_j - b_i))}\right)$ , e que  $P(Z_{ij} = 0|Y_{ij} = 0, \cdot) = 1$ . Além disso  $(X_{ij}|Z_{ij} = 1, \cdot) \sim N_{[0, \infty)}(a_i\theta_j - b_i, 1)$  e  $(X_{ij}|Z_{ij} = 0, \cdot) \sim N_{(-\infty, 0]}(a_i\theta_j - b_i, 1)$  pela construção das variáveis auxiliares.

Para o segundo bloco é atribuído que  $\theta_j \sim N(0, 1)$ , assim:

$$\pi(\theta_j|\cdot) \propto \prod_{i=1}^I \pi(X_{ij}|a_i, b_i, \theta_j)\pi(\theta_j).$$

Os cálculos, apresentados no Apêndice A.1 mostram que:

$$(\theta_j|\cdot) \sim N\left(\frac{\sum_{i=1}^I a_i(X_{ij} + b_i)}{\sum_{i=1}^I a_i^2 + 1}, \frac{1}{\sum_{i=1}^I a_i^2 + 1}\right).$$

Para o terceiro bloco,  $\Psi_3$ , são atribuídas as seguintes distribuições a priori:  $a_i, b_i$ ,  $a_i \sim N_{[0,\infty)}(\mu_a, \sigma_a^2)$ ,  $b_i \sim N(\mu_b, \sigma_b^2)$ , assim:

$$\pi(a_i, b_i | \cdot) \propto \prod_{j=1}^J \pi(X_{ij} | a_i, b_i, \theta_j) \pi(a_i) \pi(b_i),$$

e, conforme mostrado no Apêndice A.1:

$$(a_i, b_i | \cdot) \sim N_2(\mu, \Sigma) I_{a_i > 0}.$$

$$\mu = \begin{bmatrix} \mu_a^* \\ \mu_b^* \end{bmatrix}, \Sigma = \begin{bmatrix} \sigma_a^{2*} & \sigma_a^* \sigma_b^* \gamma \\ \sigma_a^* \sigma_b^* \gamma & \sigma_b^{2*} \end{bmatrix}, \text{ onde:}$$

$$\sigma_a^{2*} = \frac{\sigma_a^2}{(\sigma_a^2 \sum_{j=1}^J \theta_j^2 + 1)(1 - \gamma^2)},$$

$$\sigma_b^{2*} = \frac{\sigma_b^2}{(\sigma_b^2 J + 1)(1 - \gamma^2)},$$

$$\gamma = \frac{\sigma_a \sigma_b \sum_{j=1}^J \theta_j}{[(\sigma_a^2 \sum_{j=1}^J \theta_j^2 + 1)(\sigma_b^2 J + 1)]^{1/2}},$$

$$\mu_a^* = \sigma_a^{2*} \left( \sum_{j=1}^J X_{ij} \theta_j + \mu_a \sigma_a^{-2} \right) - \sigma_a^* \sigma_b^* \rho \left( \sum_{j=1}^J X_{ij} - \mu_b \sigma_b^{-2} \right),$$

$$\mu_b^* = \sigma_a^* \sigma_b^* \rho \left( \sum_{j=1}^J X_{ij} \theta_j + \mu_a \sigma_a^{-2} \right) - \sigma_b^{2*} \left( \sum_{j=1}^J X_{ij} - \mu_b \sigma_b^{-2} \right).$$

Para o quarto bloco é atribuído  $c_i \sim \text{Beta}(\alpha_c, \beta_c)$ :

$$\pi(c_i | \cdot) = \prod_{j=1}^J \pi(Y_{ij} | Z_{ij}, c_i) \pi(c_i),$$

o que leva a (ver Apêndice A.1):

$$(c_i | \cdot) \sim \text{Beta} \left( \sum_{j|Z_{ij}=0} Y_{ij} + \alpha_c, \sum_{j=1}^J I_{Z_{ij}=0} - \sum_{j|Z_{ij}=0} Y_{ij} + \beta_c \right).$$

---

**Algoritmo 1:** Gibbs Sampling

---

**Entrada:** Banco de dados, valor inicial para os parâmetros, valores para os hiperparâmetros (parâmetro das distribuições *a priori*), número de iterações desejadas.

**Saída:** Amostras de cada parâmetro e traço latente de tamanho.

1 **início**

2     **para**  $k$  de 1 até o número de iterações **faça**

3         Gere uma amostra de  $\psi_1$ , utilizando a distribuição condicional completa.  
           $(X_{ij}, Z_{ij}|\cdot)$ ;

4         Gere uma amostra de  $\psi_2$ , através da distribuição de  $(\theta_j|\cdot)$ ;

5         Gere uma amostra de  $\psi_3$ , através da distribuição de  $(a_i, b_i|\cdot)$ ;

6         Gere uma amostra de  $\psi_4$ , através da distribuição de  $(c_i|\cdot)$ ;

7     **fim**

8 **fim**

---

## 2.2 Algoritmo Gonçalves et al. (2018)

O algoritmo proposto por Gonçalves et al. (2018), utiliza-se também de duas variáveis auxiliares para a construção do algoritmo Gibbs-Samplings que, assim como Béguin and Glas (2001), também permite amostrar diretamente das distribuições condicionais completas. Os autores consideram uma variável auxiliar  $Z_{ij} \sim Bernoulli(c_i)$  e  $X_{ij}$ , com  $(X_{ij}|Z_{ij} = 0) \sim N(a_i\theta_j - b_i, 1)$  e  $(X_{ij}|Z_{ij} = 1)$  é um ponto de massa em zero, isto é,  $P(X_{ij} = 0|Z_{ij} = 1) = 1$ . Assim a equação (1) pode ser escrita da seguinte maneira:

$$Y_{ij} = \begin{cases} 1 & \text{se } Z_{ij} = 1 \text{ ou } (X_{ij} > 0, Z_{ij} = 0) \\ 0 & \text{se } (X_{ij} < 0, Z_{ij} = 0) \end{cases}$$

Pode-se perceber que, desta maneira, é preservada a distribuição marginal em (2):

$$\begin{aligned} P(Y_{ij} = 1|a_i, b_i, c_i, \theta_j) &= P(X_{ij} = 0|Z_{ij} = 1)P(Z_{ij} = 1) + P(X_{ij} > 0|Z_{ij} = 0)P(Z_{ij} = 0) \\ &= c_i + (1 - c_i) \int_0^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(X_{ij} - a_i\theta_j + b_i)^2} \partial X_{ij} \\ &= c_i + (1 - c_i)\Phi(a_i\theta_j - b_i) \end{aligned}$$

Tem-se então  $\Psi = \{\mathbf{Z}, \mathbf{X}, \mathbf{a}, \mathbf{b}, \mathbf{c}, \theta\}$ , onde  $\mathbf{c} = (c_1, \dots, c_I)$ ,  $\mathbf{a} = (a_1, \dots, a_I)$ ,  $\mathbf{b} = (b_1, \dots, b_I)$ ,  $\theta = (\theta_1, \dots, \theta_J)$ ,  $\mathbf{X}, \mathbf{Z}$  são matrizes de tamanho  $I \times J$  com entradas  $X_{ij}$  e  $Z_{ij}$  com  $i=1, \dots, I$  e



$j=1, \dots, J$ . A blocagem para a simulação deste algoritmo é a mesma utilizada para o algoritmo de Béguin and Glas (2001).

Para o bloco  $\psi_1 = \{Z, X\}$ , tem-se que:

$$\pi(Z_{ij}, X_{ij} | \cdot) \propto \pi(Y_{ij} | Z_{ij}, X_{ij}) \pi(Z_{ij} | c_i) \pi(X_{ij} | a_i, b_i, \theta_j).$$

Os cálculos apresentados em Apêndice A.2 mostram que, quando  $Y_{ij} = 0$ , tem-se que  $Z_{ij}$  é um ponto de massa em zero e  $X_{ij}$  é uma Normal com média  $a_i \theta_j - b_i$  truncada nos valores negativos. Para  $Y_{ij} = 1$ ,  $Z_{ij} \sim \text{Bernoulli}(w_{ij})$  e  $X_{ij}$  é um ponto de massa em zero se  $Z_{ij} = 1$  e  $X_{ij} \sim N_{(0, \infty)}(a_i \theta_j - b_i, 1)$ , se  $Z_{ij} = 0$ , para  $w_{ij} = \frac{c_i}{c_i + (1 - c_i) \Phi(a_i \theta_j - b_i)}$ .

Para o segundo bloco, utilizando uma *priori* Normal(0,1) para  $\theta_j$ , tem-se que:

$$\pi(\theta_j | \cdot) \propto \prod_{i; Z_{ij}=0} \pi(X_{ij} | a_i, \theta_j, b_i, Z_{ij}) \pi(\theta_j | \cdot),$$

Os cálculos, apresentados no Apêndice A.2, mostram que:

$$(\theta_j | \cdot) \sim N \left( \frac{\sum_{i; Z_{ij}=0} a_i (X_{ij} - b_i)}{\sum_{i; Z_{ij}=0} a_i^2 + 1}, \frac{1}{\sum_{i; Z_{ij}=0} a_i^2 + 1} \right).$$

Para o terceiro bloco, utilizando a mesma *priori* que o algoritmo de Béguin e Glas, tem-se que a condicional completa é:

$$(a_i, b_i | \cdot) \sim N_2(\mu, \Sigma) I_{a_i > 0}$$

$$\mu = \begin{bmatrix} \mu_a^* \\ \mu_b^* \end{bmatrix}, \Sigma = \begin{bmatrix} \sigma_a^{2*} & \sigma_a^* \sigma_b^* \gamma \\ \sigma_a^* \sigma_b^* \gamma & \sigma_b^{2*} \end{bmatrix}, \text{ onde:}$$

$$\sigma_a^{2*} = \frac{\sigma_a^2}{(\sigma_a^2 \sum_{j; Z_{ij}=0} \theta_j^2 + 1)(1 - \gamma^2)},$$

$$\sigma_b^{2*} = \frac{\sigma_b^2}{(\sigma_b^2 N_J + 1)(1 - \gamma^2)},$$

com  $N_J = \sum_{j=1}^J I_{\{Z_{ij}=0\}}$ .

$$\gamma = \frac{\sigma_a \sigma_b \sum_{j; Z_{ij}=0} \theta_j}{[(\sigma_a^2 \sum_{j; Z_{ij}=0} \theta_j^2 + 1)(\sigma_b^2 N_J + 1)]^{1/2}},$$

$$\mu_a^* = \sigma_a^{2*} \left( \sum_{j; Z_{ij}=0} X_{ij} \theta_j + \mu_a \sigma_a^{-2} \right) - \sigma_a^* \sigma_b^* \gamma \left( \sum_{j; Z_{ij}=0} X_{ij} - \mu_b \sigma_b^{-2} \right),$$

$$\mu_b^* = \sigma_a^* \sigma_b^* \gamma \left( \sum_{j; Z_{ij}=0} X_{ij} \theta_j + \mu_a \sigma_a^{-2} \right) - \sigma_b^{2*} \left( \sum_{j; Z_{ij}=0} X_{ij} - \mu_b \sigma_b^{-2} \right).$$

Para o quarto bloco, utilizando a *priori* que  $c_i \sim \text{Beta}(\alpha_c, \beta_c)$ , tem-se que:

$$\pi(c_i | \cdot) \propto \prod_{j=1}^J \pi(Z_{ij} | c_i) \pi(c_i),$$

que leva a após alguns cálculos (ver Apêndice A.2):

$$c_i \sim \text{Beta} \left( \sum_{j=1}^J Z_{ij} + \alpha_c, J - \sum_{j=1}^J Z_{ij} + \beta_c \right).$$

O algoritmo é então estruturado da mesma maneira que o algoritmo apresentado em Algoritmo 1, porém são utilizadas as condicionais completas mostradas nesta seção.

## 2.3 Algoritmo Metropolis-Hastings I

Nesta subseção iremos apresentar uma variação do algoritmo proposto por (Gonçalves et al., 2018). Para este algoritmo será retirada a variável auxiliar  $X$ . Assim as coordenadas das cadeias de Markov são  $\Psi = \{\mathbf{Z}, \mathbf{a}, \mathbf{b}, \mathbf{c}, \theta\}$  e

$$P(Y_{ij} = 1) = \begin{cases} 1 & ; \text{se } Z_{ij} = 1 \\ \Phi(a_i \theta_j + b_i) & ; \text{se } Z_{ij} = 0. \end{cases}$$

Com a retirada da variável auxiliar  $X$ , não é possível amostrar diretamente das condi-

cionais completas dos parâmetros  $\mathbf{a}, \mathbf{b}$ , logo, para estes parâmetros será utilizado um passo Metropolis-Hastings. Para os traços latentes é possível se amostrar diretamente, utilizando a distribuição skew-normal, porém, devido ao alto custo computacional para fazê-lo, será considerado um passo Metropolis-Hastings. Os blocos serão os mesmos que os anteriores, exceto por  $\psi_1 = \{Z\}$ .

Para o primeiro bloco temos que a condicional completa é:

$$(Z_{ij}|Y_{ij} = 1, \cdot) \sim \text{Bernoulli} \left( \frac{c_i}{c_i + (1 - c_i)\Phi(a_i\theta_j - b_i)} \right).$$

Quando  $Y_{ij} = 0$  temos que  $Z_{ij}$  é um ponto de massa em zero (ver Apêndice A.3). O segundo bloco não é amostrado diretamente da sua condicional completa, logo é utilizado um passo Metropolis-Hastings.

Assim, no passo k, tem-se que o valor proposto para  $\theta_j^k$  é gerado de uma distribuição Normal( $\theta_j^{k-1}, \tau_{\theta_j}$ ). Considerando como distribuição *priori* para  $\theta_j$  distribuição Normal padrão, temos que a probabilidade de aceitação na iteração k é:

$$\alpha(\theta_j^{k-1}, \theta_j^*) = \min \left( 1, \exp\{-0.5(\theta_j^{*2} - \theta_j^{k-12})\} \prod_{i; Z_{ij}=0} \left[ \frac{p_i(\theta_j^*)}{p_i(\theta_j^{k-1})} \right]^{Y_{ij}} \left[ \frac{1 - p_i(\theta_j^*)}{1 - p_i(\theta_j^{k-1})} \right]^{1 - Y_{ij}} \right),$$

onde  $\theta_j^*$  é o valor proposto e  $p_i(\theta) = \Phi(a_i\theta - b_i)$ .

Para o terceiro bloco é utilizado um passo Metropolis-Hastings. A distribuição proposta é uma Normal Bivariada, com correlação 0.

Assim, na iteração k, o valor proposto para  $a_i$  é gerado de uma distribuição Normal( $a_i^{k-1}, \tau_{a_i}$ ) e  $b_i$  é gerado de uma distribuição Normal( $b_i^{k-1}, \tau_{b_i}$ ). Considerando as distribuições a *priori* utilizadas anteriormente para a e b, temos que a probabilidade de aceitação é:

$$\alpha((a_i^{k-1}, b_i^{k-1}), (a_i^*, b_i^*)) = \min \left( 1, \exp \left\{ -\frac{1}{2\sigma_a^2} (a_i^{*2} - a_i^{k-12} - 2\mu_a(a_i^* - a_i^{k-1})) \right\} \exp \left\{ -\frac{1}{2\sigma_b^2} (b_i^{*2} - b_i^{k-12} - 2\mu_b(b_i^* - b_i^{k-1})) \right\} \prod_{j; Z_{ij}=0} \left[ \frac{p_j(a_i^*, b_i^*)}{p_j(a_i^{k-1}, b_i^{k-1})} \right]^{Y_{ij}} \left[ \frac{1 - p_j(a_i^*, b_i^*)}{1 - p_j(a_i^{k-1}, b_i^{k-1})} \right]^{1 - Y_{ij}} I_{a_i^* > 0} \right),$$

onde  $a_i^*$  e  $b_i^*$  são valores propostos e  $p_j(a, b) = \Phi(a\theta_j - b)$ .

Para o quarto bloco temos a mesma distribuição condicional completa do algoritmo de Gonçalves et al. (2018), assim:

$$c_i \sim \text{Beta} \left( \sum_{j=1}^J Z_{ij} + \alpha_c, J - \sum_{j=1}^J Z_{ij} + \beta_c \right).$$

A estrutura do algoritmo é apresentada em Algoritmo 2.

---

**Algoritmo 2:** METROPOLIS-HASTINGS I

---

**Entrada:** Banco de dados, valor inicial dos parâmetros, valores dos hiperparâmetros, número de iterações desejadas, valor inicial para a variância das distribuições de simulação.

**Saída:** Amostras de cada parâmetro e traço latente.

```

1 início
2   para  $k$  de 1 até o número de iterações faça
3     Gere uma amostra de  $\psi_1$ , utilizando a distribuição da condicional completa de
       ( $Z_{ij}|\cdot$ );
4     Gere um valor proposto de  $\psi_2$  através da distribuição  $N(\theta_j^{k-1}, \tau_{\theta_j})$ . Gere
        $B_1 \sim \text{Bernoulli}(\alpha(\theta_j^*, \theta_j^{k-1}))$ ;
5     se  $B_1 = 1$  então
6       |  $\theta_j^k = \theta_j^*$ ;
7     senão
8       |  $\theta_j^k = \theta_j^{k-1}$ ;
9     fim
10    Gere um valor proposto de  $\psi_3$  através da distribuição Normal bivariada. Gere
        $B_2 \sim \text{Bernoulli}(\alpha((a_i^*, b_i^*), (a_i^{k-1}, b_i^{k-1})))$ ;
11    se  $B_2 = 1$  então
12      |  $a_i^k = a_i^*, b_i^k = b_i^*$ ;
13    senão
14      |  $a_i^k = a_i^{k-1}, b_i^k = b_i^{k-1}$ ;
15    fim
16    Gere uma amostra de  $\psi_4$ , utilizando a distribuição da condicional completa de
       ( $c_i|\cdot$ );
17  fim
18 fim

```

---

As variâncias das distribuições propostas são alteradas até se obter as taxas de aceitação desejadas.

## 2.4 Algoritmo Metropolis-Hastings II

Nesta subseção iremos apresentar um algoritmo Metropolis-Hastings para o modelo de TRI de três parâmetros. Este algoritmo não considera nenhuma variável auxiliar e, portanto, as coordenadas das cadeias de Markov são  $\Psi = \{a, b, c, \theta\}$ .

Todos os blocos,  $\Psi_1 = \{\theta\}$ ,  $\Psi_2 = \{a, b\}$ ,  $\Psi_3 = \{c\}$  são amostrados via passo Metropolis Hasting.

Para o primeiro bloco tem-se que o valor proposto para  $\theta_j^k$  é gerado de uma distribuição Normal( $\theta_j^{k-1}, \tau_{\theta_j}$ ). Considerando que a *priori* para  $\theta_j$  é Normal(0,1), temos que a probabilidade de aceitação é:

$$\alpha(\theta_j^{k-1}, \theta_j^*) = \min \left( 1, \exp\{-0.5(\theta_j^{*2} - \theta_j^{k-12})\} \prod_{i=1}^I \left[ \frac{p_i(\theta_j^*)}{p_i(\theta_j^{k-1})} \right]^{Y_{ij}} \left[ \frac{1 - p_i(\theta_j^*)}{1 - p_i(\theta_j^{k-1})} \right]^{1-Y_{ij}} \right),$$

onde  $\theta_j^*$  é o valor proposto e  $p_i(\theta) = c_i + (1 - c_i)\Phi(a_i\theta - b_i)$ .

Para o segundo bloco tem-se que o valor proposto para  $a_i$  é gerado de uma distribuição Normal( $a_i^{k-1}, \tau_{a_i}$ ) e  $b_i$  é gerado de uma distribuição Normal( $b_i^{k-1}, \tau_{b_i}$ ). Considerando as distribuições a *priori* utilizadas anteriormente para a e b, temos que a probabilidade de aceitação é:

$$\alpha((a_i^{k-1}, b_i^{k-1}), (a_i^*, b_i^*)) = \min \left( 1, \exp \left\{ -\frac{1}{2\sigma_a^2} (a_i^{*2} - a_i^{k-12} - 2\mu_a(a_i^* - a_i^{k-1})) \right\} \exp \left\{ -\frac{1}{2\sigma_b^2} (b_i^{*2} - b_i^{k-12} - 2\mu_b(b_i^* - b_i^{k-1})) \right\} \prod_{j=1}^J \left[ \frac{p_{ij}(a_i^*, b_i^*)}{p_{ij}(a_i^{k-1}, b_i^{k-1})} \right]^{Y_{ij}} \left[ \frac{1 - p_{ij}(a_i^*, b_i^*)}{1 - p_{ij}(a_i^{k-1}, b_i^{k-1})} \right]^{1-Y_{ij}} I_{a_i^* > 0} \right).$$

Para o terceiro bloco, o valor proposto para  $c_i$  é gerado de uma distribuição Uniforme ( $\psi(c_i^{k-1}), \gamma(c_i^{k-1})$ ), onde  $\psi(c_i^{k-1}) = \max(0, c_i^{k-1} - \delta)$  e  $\gamma(c_i^{k-1}) = \min(1, c_i^{k-1} + \delta)$ . Utilizando

a distribuição a *priori* utilizada anteriormente, tem-se que:

$$\alpha(c_i^{k-1}, c_i^*) = \min \left( 1, \left( \frac{c_i^*}{c_i^{k-1}} \right)^{\alpha_{c-1}} \left( \frac{1 - c_i^*}{1 - c_i^{k-1}} \right)^{\beta_{c-1}} \prod_{j=1}^J \left[ \frac{p_{ij}(c_i^*)}{p_{ij}(c_i^{k-1})} \right]^{Y_{ij}} \left[ \frac{1 - p_{ij}(c_i^*)}{1 - p_{ij}(c_i^{k-1})} \right]^{1 - Y_{ij}} \frac{\gamma(c_i^*) - \psi(c_i^*)}{\gamma(c_i^{k-1}) - \psi(c_i^{k-1})} \right).$$

A estrutura do algoritmo é apresentada no Algoritmo 3.

---

**Algoritmo 3:** METROPOLIS-HASTINGS II

---

**Entrada:** Banco de dados, valor inicial dos parâmetros, valores dos hiperparâmetros, número de iterações desejadas, valor inicial para a variância das distribuições de simulação.

**Saída:** Amostras de cada parâmetro e traço latente.

1 **início**

2     **para**  $k$  de 1 até o número de iterações **faça**

3         Gere um valor proposto de  $\psi_1$  através da distribuição  $N(\theta_j^{k-1}, \tau_{\theta_j})$ . Gere  
         $B_1 \sim \text{Bernoulli}(\alpha(\theta_j^*, \theta_j^{k-1}))$ ;

4         **se**  $B_1 = 1$  **então**

5              $\theta_j^k = \theta_j^*$ ;

6         **senão**

7              $\theta_j^k = \theta_j^{k-1}$ ;

8         **fim**

9         Gere um valor proposto de  $\psi_2$  através da distribuição Normal bivariada. Gere

$B_2 \sim \text{Bernoulli}(\alpha((a_i^*, b_i^*), (a_i^{k-1}, b_i^{k-1})))$ ;

10        **se**  $B_2 = 1$  **então**

11            $a_i^k = a_i^*, b_i^k = b_i^*$ ;

12        **senão**

13            $a_i^k = a_i^{k-1}, b_i^k = b_i^{k-1}$ ;

14        **fim**

15        Gere uma amostra de  $\psi_3$ , através da distribuição Uniforme( $\psi(c_i^{k-1}), \gamma(c_i^{k-1})$ );

       Gere  $B_3 \sim \text{Bernoulli}(\alpha(c_i^*, c_i^{k-1}))$ ;

16        **se**  $B_3 = 1$  **então**

17            $c_i^k = c_i^*$ ;

18        **senão**

19            $c_i^k = c_i^{k-1}$ ;

20        **fim**

21     **fim**

22 **fim**

---

As variâncias das distribuições propostas para os parâmetros  $a$ ,  $b$  e  $\theta$  e os limites para a distribuição proposta para o parâmetro  $c$  são calibradas para se obter as taxas de aceitação desejadas.

## 2.5 Critérios para comparação

Ambos, Béguin and Glas (2001) e Gonçalves et al. (2018) propõem um algoritmo Gibbs Samplings com variáveis auxiliares que permitem a amostragem direta de todas as condicionais completas. Uma das variáveis auxiliares é comum aos dois algoritmos, a diferença se faz na variável auxiliar utilizada para resolver o problema de intratabilidade causada pelo parâmetro de acerto casual. Enquanto a variável proposta por Béguin and Glas (2001) é dependente dos parâmetros de discriminação, dificuldade e traço latente, a variável proposta por Gonçalves et al. (2018) depende apenas do parâmetro de acerto casual. Isto sugere que a autocorrelação introduzida na cadeia por essas variáveis será menor para o algoritmo de Gonçalves et al. (2018).

Os algoritmos Metropolis-Hastings não utilizam variável auxiliar e, portanto, não introduzem este tipo de autocorrelação. Porém, introduzem a autocorrelação associada a proposta de passeio aleatório.

Para comparação dos algoritmos foram armazenadas as cadeias, de cada parâmetro dos itens, para os itens que possuíam maior discriminação, ou maior dificuldade, ou maior acerto casual, e itens que possuíam maior e menor proporção de acerto. No total foram armazenados doze cadeias de itens para análises. Para as cadeias dos traços latentes foram armazenadas quatro cadeias, uma de maior valor de traço latente, uma de menor valor de traço latente e duas de valores próximos a 0. Também foi armazenada a cadeia do logaritmo da função de densidade *posteriori*, isto é:

$$\begin{aligned} \log(\pi(\theta, a, b, c|Y)) &\propto \log(\pi(Y|\theta, a, b, c)\pi(\theta)\pi(a)\pi(b)\pi(c)) \\ &\propto \sum_{i=1}^I \sum_{j=1}^J \log(\pi(Y_{ij}|\theta_j, a_i, b_i, c_i)) \\ &+ \sum_{j=1}^J \log(\pi(\theta_j)) + \sum_{i=1}^I \log(\pi(a_i)) + \log(\pi(b_i)) + \log(\pi(c_i)). \end{aligned} \tag{6}$$

Os primeiros critérios de comparação dos algoritmos são: o tamanho do burn-in, o tempo computacional médio até este momento de burn-in, e um tempo médio de cada iteração.

Para a escolha do tamanho de burn-in foi utilizada a estatística proposta por Gelman et al. (1992). Para esta estatística são necessárias múltiplas cadeias começando em pontos diferentes. Esta estatística analisa a convergência comparando a variância entre cadeias e dentro das cadeias. A sua estimativa deve ser próxima de 1 para indicativo de convergência. Para a definição de burn-in calculou-se esta estatística utilizando diferentes valores de burn-in e o valor mais baixo foi o valor selecionado.

Um dos principais comparativos dos algoritmos é a função de autocorrelação (FAC) da cadeia. Como o modelo possui diferentes parâmetros foi utilizada a cadeia da função log da densidade conjunta como uma maneira de sintetizar as informações de todos os parâmetros.

Supondo que há  $N$  iterações de uma cadeia univariada de  $\Theta$ , para se calcular a FAC de ordem  $k$ , tem-se a seguinte equação:

$$r_k = \frac{\sum_{i=1}^{N-k} (\Theta_i - \bar{\Theta})(\Theta_{i+k} - \bar{\Theta})}{\sum_{i=1}^N (\Theta_i - \bar{\Theta})^2}, \quad (7)$$

onde  $\bar{\Theta}$  é média da cadeia. Neste trabalho é calculada a função de autocorrelação após o burn-in. Quanto menor este valor, menos correlacionada estará a amostra, assim o algoritmo tem um desempenho melhor, em que os respectivos estimadores de Monte Carlo terão menor variância.

O último critério adotado neste trabalho é a utilização do tamanho efetivo de amostra, calculado através de:

$$TEA = \frac{N}{1 + 2 \sum_{k=1}^{\infty} r_k}. \quad (8)$$

O tamanho efetivo de amostra representa o número de amostras, independentes, que deveriam ser retiradas da distribuição limite do MCMC para que se obtivesse um estimador de Monte Carlo com variância idêntica ao obtido pela amostra, de tamanho  $N$ , gerada através do MCMC, assim, quanto maior o tamanho efetivo de amostra melhor o estimador. O fator de ineficiência,  $1 + 2 \sum_{k=1}^{\infty} r_k$ , é um indicativo do quanto o algoritmo é eficaz, uma vez que é denominador do tamanho efetivo de amostra, quanto maior o seu valor menor será o tamanho



efetivo de amostra.

Para comparação calculou-se o tamanho efetivo de amostra para um valor de  $N$  fixo, e assim calculada a razão entre os tamanhos efetivos e tempos computacionais de dois algoritmos concorrentes.

Também foi calculada uma estimativa de número de iterações necessárias em cada algoritmo para que se obtivesse um tamanho efetivo de amostra fixo. Desta forma, foi-se comparada a repetição de melhor desempenho, repetição que se obteve o valor de maneira mais rápida, de cada algoritmo fixando um tempo computacional e comparado o tamanho efetivo de amostra de cada algoritmo, pois um algoritmo pode obter um número menor de iterações neste tempo fixo, porém um maior tamanho efetivo de amostra.

### 3 Simulações

Nesta seção serão apresentados os estudos simulados para comparação dos quatro algoritmos de MCMC apresentados na Seção 2. Para investigar a influência de tamanho de amostra, são considerados diferentes conjuntos de dados variando o número de itens e de indivíduos. Todas as análises consideram as seguintes distribuições *a priori*:  $a_i \sim N(1, 3)$ ,  $b_i \sim N(0, 4)$ ,  $c_i \sim Beta(4, 12)$ ,  $\theta_j \sim N(0, 1)$ .

Três conjuntos de dados simulados de diferentes tamanhos são considerados para as análises, conforme mostrado na Tabela 1. Para todos os três conjunto de dados, os parâmetros de discriminação são gerados de uma Uniforme (0.8,2.2), a dificuldade de uma Uniforme (-3,3), o parâmetro de acerto casual de uma Uniforme (0,0.2) e os traços latentes uma distribuição Normal (0,1).

	Número de itens	Número de indivíduos
1	20	500
2	40	2000
3	60	10000

Tabela 1: Número de itens e indivíduos por simulação

São realizadas quatro repetições de cada algoritmo, iniciando a cadeia de pontos diferentes, em cada um dos estudos, para auxiliar a escolha do valor de burn-in. Para todas as simulações foram realizadas 150000 iterações da cadeia de Markov.

Em todas as repetições foram atribuídos valores iniciais de 0.1 para o parâmetro de acerto casual e para o traço latente foi calculado o escore bruto padronizado de cada indivíduo, dado por:

$$EBP_j = \frac{s_j - \bar{s}}{\sqrt{Var(s)}}, \quad (9)$$

$$s_j = \sum_{i=1}^I Y_{ij},$$

$$\bar{s} = \frac{\sum_{j=1}^J s_j}{J},$$

$$Var(s) = \frac{\sum_{j=1}^J (s_j - \bar{s})^2}{J - 1}.$$

As quatro repetições se diferenciam com relação aos valores iniciais dos parâmetros  $a_i$  e  $b_i$  conforme Tabela 2.

	$\theta$	$a$	$b$	$c$
Repetição 1	$EBP_j$	1	0	0.1
Repetição 2	$EBP_j$	1	$EBP_i$	0.1
Repetição 3	$EBP_j$	1.5	0	0.1
Repetição 4	$EBP_j$	1.5	$1.5EBP_i$	0.1

Tabela 2: Valores iniciais de cada parâmetro em cada repetição

Para o parâmetro  $b$  foi utilizado o escore bruto padronizado de cada item.

$$EBP_i = \frac{s_i - \bar{s}}{\sqrt{Var(s)}}, \quad (10)$$

$$s_i = \sum_{j=1}^J Y_{ij},$$

$$\bar{s} = \frac{\sum_{i=1}^I s_i}{I},$$

$$Var(s) = \frac{\sum_{i=1}^I (s_i - \bar{s})^2}{I - 1}.$$

Baseado no comportamento da cadeia da log-densidade a *posteriori* (ver Apêndice C) e de alguns parâmetros, escolheu-se utilizar um burn-in de 10 mil iterações para todos os exemplos.

### 3.1 Estudo simulado 1

Nesta seção serão apresentados os resultados para o cenário simulado com 500 indivíduos e 20 itens.

A Tabela 3 apresenta o tempo médio em minutos até se obter o valor de Burn-in (10000 iterações) e o tempo médio, em segundos, de cada iterações.

A Figura 6 apresenta gráficos comparativos, dois a dois, da função de autocorrelação (FAC) da cadeia log-densidade a *posteriori* de cada algoritmo para a primeira repetição. Percebe-se que o algoritmo de Béguin and Glas (2001) apresenta maior valor de autocorrelação que os outros três algoritmos. Esses, por sua vez, têm comportamentos parecidos,

	Béguin and Glas (2001)	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Tempo médio até o burn-in	1:35.33	1:44.75	2:00.05	2:14.25
Tempo médio por iteração	0.016s	0.017s	0.018s	0.019s

Tabela 3: Tempo computacional, médio, por iteração e até se obter o burn-in por algoritmo para estudo simulado 1.

porém os algoritmos Metropolis apresentam valores baixos de autocorrelação para ordens superiores. O fator de ineficiência de cada um dos quatro algoritmos é 48.58, 41.83, 41.15 e 33.7, seguindo a ordenação das colunas da Tabela 3.

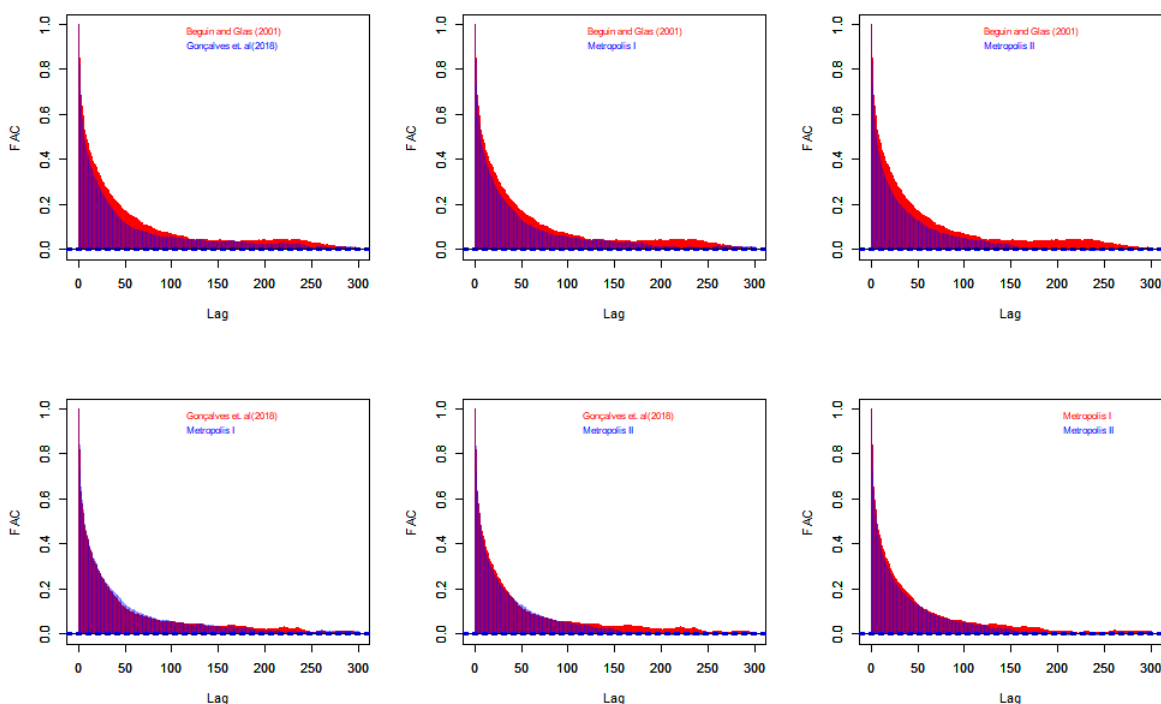


Figura 6: Comparação da função de autocorrelação da log-densidade a posteriori entre os quatro algoritmos no Estudo Simulado 1.

A Tabela 4 mostra o valor calculado para o tamanho efetivo de amostra, da função log densidade a *posteriori*, de cada uma das repetições dos algoritmos considerando um valor de burn-in 10000. Este valor mantém fixo o numerador apresentado na equação (8) para o cálculo de tamanho efetivo de amostra, assim é possível perceber o impacto do fator de ineficiência.

Percebe-se que o algoritmo Metropolis II apresenta maior tamanho efetivo de amostra em todas as repetições. O algoritmo de Béguin and Glas (2001) apresenta os menores valores

	Béguin and Glas (2001)	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Repetição 1	3040	3804	3457	3815
Repetição 2	3126	3921	3583	4115
Repetição 3	3122	2875	3255	4741
Repetição 4	2720	3434	3580	4476
Média	3002	3508	3468	4286

Tabela 4: Tamanho Efetivo de Amostra após 10000 iterações para cada repetição no Estudo Simulado 1.

de tamanho efetivo. No geral, os algoritmos Metropolis apresentam maiores tamanhos efetivos de amostra em comparação com os existentes na literatura. Os algoritmos de Béguin and Glas (2001), de Gonçalves et al. (2018) e Metropolis I apresentam maiores resultados de tamanho efetivo de amostra para a segunda repetição, já o algoritmo Metropolis II apresenta maior para a terceira repetição.

A Tabela 5 apresenta a razão entre o tempo médio por iteração de cada algoritmo. A sua leitura se dá pela divisão do tempo médio do algoritmo referente da coluna, pelo algoritmo da linha, exemplo, algoritmo de Gonçalves et al. (2018) apresenta tempo médio 1.08 vezes o tempo médio por iteração para o algoritmo de Béguin and Glas (2001).

	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Béguin and Glas (2001)	1.08	1.15	1.21
Gonçalves et al. (2018)		1.07	1.12
Metropolis I			1.05

Tabela 5: Razão entre os tempos médios por iteração de cada algoritmo para Estudo Simulado 1.

A Tabela 6 apresenta a razão entre o valor médio do tamanho efetivo de amostra de cada uma das repetições, sua leitura se dá de maneira similar a leitura da Tabela 5.

	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Béguin and Glas (2001)	1.17	1.16	1.43
Gonçalves et al. (2018)		0.99	1.22
Metropolis I			1.24

Tabela 6: Razão entre o tamanho efetivo de amostra médio de cada algoritmo para Estudo Simulado 1.

É possível perceber através da Tabela 5 e Tabela 6 que o algoritmo de Gonçalves et al. (2018) apresenta um tempo médio por iteração 1.08 vezes o tempo médio por iteração do

algoritmo de Béguin and Glas (2001), porém apresenta um tamanho efetivo de amostra 1.17 vezes o tamanho efetivo de amostra do algoritmo de Béguin and Glas (2001).

Logo há evidências de que o algoritmo de Gonçalves et al. (2018) demanda um tempo computacional menor que o algoritmo de Béguin and Glas (2001) para se obter um mesmo tamanho efetivo de amostra, uma vez que, em média, o algoritmo de Gonçalves et al. (2018) obtém maiores tamanhos efetivos de amostra com um baixo custo computacional a mais. Os resultados sugerem a seguinte ordenação decrescente dos algoritmos em termos de eficiência computacional: Metropolis II, Gonçalves et al. (2018), Metropolis I, Béguin and Glas (2001).

Este resultado, porém, considera o desempenho médio nas quatro repetições. É razoável considerar apenas a repetição com a melhor performance para cada um dos algoritmos. Para tal, a Tabela 7 apresenta o tempo, em minutos, necessário para cada algoritmo obter um tamanho efetivo de amostra fixo (2720). Quando consideramos a repetição de melhor desempenho de cada algoritmo, temos a seguinte ordenação decrescente por eficiência: Gonçalves et al. (2018), Metropolis II, Metropolis I, Béguin and Glas (2001).

	Béguin and Glas (2001)		Gonçalves et al. (2018)		Metropolis I		Metropolis II	
	Nº de iterações	Tempo	Nº de iterações	Tempo	Nº de iterações	Tempo	Nº de iterações	Tempo
Repetição 1	134614	33:10.45	109472	27:01.28	123755	34:41.22	105602	29:52.42
Repetição 2	132244	32:24.26	101241	24:42.24	109075	29:23.78	105452	29:55.41
Repetição 3	130359	32:01.46	142971	38:38.92	124098	34:49.45	102494	28:45.73
Repetição 4	150000	38:44.26	121234	31:07.82	109713	29:42.56	93006	25:33.05

Tabela 7: Número de iterações e tempo computacional necessário para se obter um tamanho efetivo de amostra fixo para Estudo Simulado 1.

Para completarmos este estudo, a Tabela 8 e Tabela 9 apresentam a razão entre os tempos computacionais apresentados na repetição de melhor desempenho de cada algoritmo e a razão de tamanho efetivo de amostra da repetição. E a Tabela 10 mostra o números de iterações e tamanho efetivo de amostra, das repetições de melhor desempenho, para um tempo computacional fixo.

	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Béguin and Glas (2001)	1.08	1.14	1.20
Gonçalves et al. (2018)		1.06	1.11
Metropolis I			1.05

Tabela 8: Razão entre os tempos por iteração das repetições de melhor desempenho de cada algoritmo para Estudo Simulado 1.

	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Béguin and Glas (2001)	1.26	1.15	1.43
Gonçalves et al. (2018)		0.91	1.14
Metropolis I			1.25

Tabela 9: Razão entre o tamanho efetivo de amostra das repetições de melhor desempenho de cada algoritmo para Estudo Simulado 1.

	Béguin and Glas (2001)	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Total de iterações	150000	142035	135595	130202
Tamanho Efetivo de Amostra	3122	3851	3335	3838

Tabela 10: Número de iterações e tamanho efetivo de amostra, da repetição de melhor desempenho, com tempo computacional fixo para Estudo Simulado 1.

## 3.2 Estudo simulado 2

Nesta seção serão apresentados os resultados para o cenário simulado com 40 itens e 2000 indivíduos.

A Tabela 11 apresenta o tempo médio, em minutos, até se obter 10000 iterações (valor definido como burn-in) e o tempo médio, em segundos, por iteração.

	Béguin and Glas (2001)	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Tempo médio até o burn-in	11:26.00	12:08.75	14:35.25	16:17.00
Tempo médio por iteração	0.071s	0.075s	0.09s	0.099s

Tabela 11: Tempo computacional, médio, por iteração e até se obter o burn-in por algoritmo para estudo simulado 2

A Figura 7 mostra gráficos comparativos, dois a dois, da função de autocorrelação da cadeia da log densidade a *posteriori* de cada algoritmo para a primeira repetição. Onde é possível perceber que o algoritmo de Béguin and Glas (2001) apresenta maior FAC, algoritmo de Gonçalves et al. (2018) e os algoritmos Metropolis I e Metropolis II apresentam comportamentos similares. O fator de ineficiência de cada um dos quatro algoritmos é 55.39, 48.12, 58.22 e 45.6, seguindo a ordenação das colunas da 11.

A Tabela 12 mostra o valor calculado de tamanho efetivo da amostra, da função log densidade *posteriori*, de cada uma das repetições para cada algoritmo considerando um burn-in de 10000 iterações para todos os algoritmos.

Percebe-se que os algoritmos de Béguin and Glas (2001) e Metropolis I apresentam menores tamanhos efetivos de amostra, este resultado é esperado, pois estes algoritmos apre-

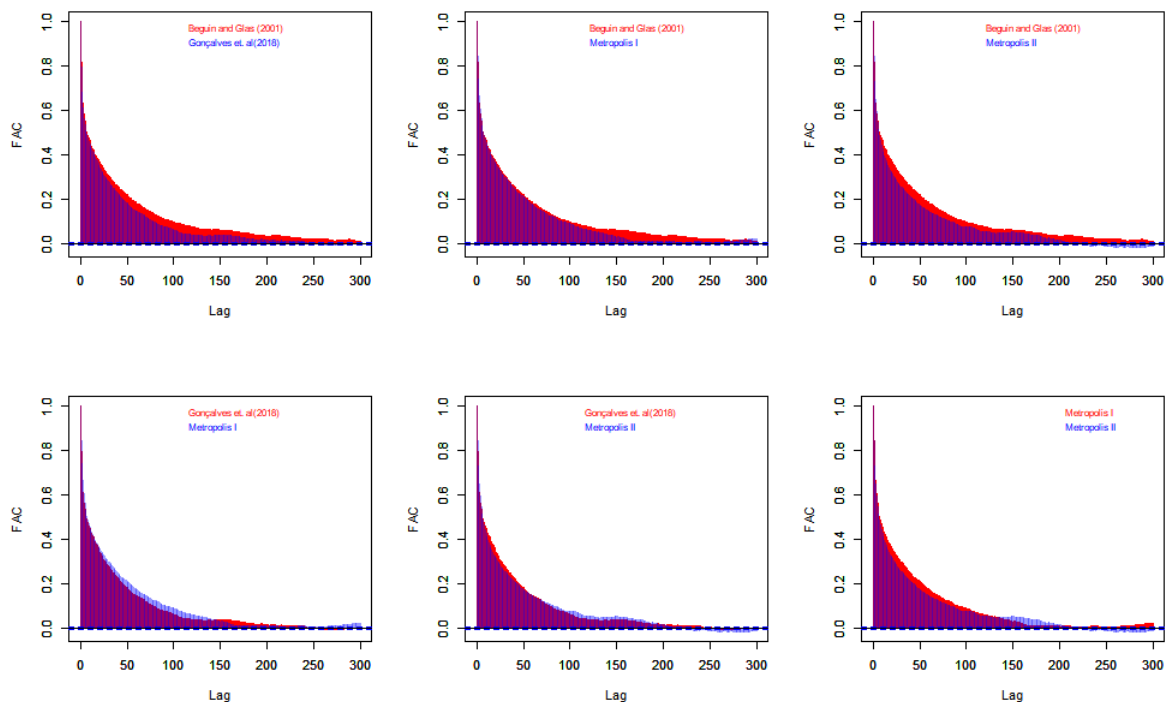


Figura 7: Comparação da função de autocorrelação da log-densidade a posteriori entre os quatro algoritmos no Estudo Simulado 2.

	Béguin and Glas (2001)	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Repetição 1	2468	2863	2595	2998
Repetição 2	2524	2748	2680	3013
Repetição 3	2615	2826	2121	2931
Repetição 4	2499	2822	2231	3133
Média	2526	2814	2406	3016

Tabela 12: Tamanho Efetivo de Amostra após 10000 iterações para cada repetição no Estudo Simulado 2.

sentaram maiores fatores de ineficiência. O algoritmo Metropolis II apresenta maior tamanho efetivo de amostra, resultado esperado uma vez que seu fator de ineficiência é menor. O algoritmo de Béguin and Glas (2001) apresenta melhor resultado para a terceira repetição, Gonçalves et al. (2018) apresenta melhor resultado na primeira repetição, Metropolis I para a segunda repetição e Metropolis II para a quarta repetição.

A Tabela 13 apresenta a razão entre o tempo médio por iteração de cada algoritmo e Tabela 14 apresenta a razão entre o valor médio do tamanho efetivo de amostra de cada uma das repetições, as leituras destas tabelas são similares as Tabela 5 e 6.



	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Béguin and Glas (2001)	1.06	1.27	1.39
Gonçalves et al. (2018)		1.19	1.31
Metropolis I			1.10

Tabela 13: Razão entre os tempos médios por iteração de cada algoritmo para Estudo Simulado 2.

	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Béguin and Glas (2001)	1.11	0.95	1.20
Gonçalves et al. (2018)		0.86	1.07
Metropolis I			1.27

Tabela 14: Razão entre o tamanho efetivo de amostra médio de cada algoritmo para Estudo Simulado 2.

É notável que o algoritmo Metropolis I apresenta baixa performance. Algoritmo Metropolis II apresenta um valor incremental para tamanho efetivo de amostra, porém seu custo computacional é maior que este incremental (exceto em comparação ao algoritmo Metropolis I). Algoritmo de Gonçalves et al. (2018) apresenta uma melhor performance em relação ao algoritmo de Béguin and Glas (2001), pois mesmo apresentando um custo computacional a mais, este apresenta um incremental ao tamanho efetivo de amostra que sobrepõe este custo. Assim, os resultados em eficiência computacional segue a seguinte ordem: Gonçalves et al. (2018), Béguin and Glas (2001), Metropolis II e Metropolis I.

Porém, estes resultados levam em consideração o valor médio. É mais razoável considerarmos o valor de melhor performance. Para isto, a Tabela 15 apresenta o tempo necessário em cada algoritmo para se obter um tamanho efetivo de amostra de 2121. Quando consideramos a melhor performance temos a seguinte ordenação: Gonçalves et al. (2018), Béguin and Glas (2001), Metropolis II e Metropolis I.

	Béguin and Glas (2001)		Gonçalves et al. (2018)		Metropolis I		Metropolis II	
	Nº de iterações	Tempo	Nº de iterações	Tempo	Nº de iterações	Tempo	Nº de iterações	Tempo
Repetição 1	121785	2:21:35	116815	2:24:24	130734	3:22:12	111612	3:00:49
Repetição 2	130549	2:32:43	113438	2:20:40	124702	3:02:58	113827	3:03:59
Repetição 3	123985	2:24:51	114235	2:20:56	150000	3:41:59	107334	2:53:00
Repetição 4	129549	2:30:45	114659	2:21:09	134151	3:16:46	108544	2:55:44

Tabela 15: Número de iterações e tempo computacional necessário para se obter um tamanho efetivo de amostra fixo para Estudo Simulado 2.

Para complementarmos os estudos, as Tabelas 16, 17 e 18 apresentam, respectivamente,

a razão entre os tempos computacionais, a razão de tamanho efetivo de amostra da repetição, o números de iterações e tamanho efetivo de amostra para um tempo computacional fixo, das repetições de melhor desempenho de cada algoritmo.

	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Béguin and Glas (2001)	1.07	1.25	1.38
Gonçalves et al. (2018)		1.17	1.29
Metropolis I			1.1

Tabela 16: Razão entre os tempos por iteração das repetições de melhor desempenho de cada algoritmo para Estudo Simulado 2.

	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Béguin and Glas (2001)	1.13	1.09	1.19
Gonçalves et al. (2018)		0.98	1.07
Metropolis I			1.09

Tabela 17: Razão entre o tamanho efetivo de amostra das repetições de melhor desempenho de cada algoritmo para Estudo Simulado 2.

	Béguin and Glas (2001)	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Total de iterações	150000	140835	120769	109683
Tamanho Efetivo de Amostra	2468	2632	2040	2161

Tabela 18: Número de iterações e tamanho efetivo de amostra, da repetição de melhor desempenho, com tempo computacional fixo para Estudo Simulado 2.

### 3.3 Estudo simulado 3

Nesta seção serão apresentados os resultados obtidos para o terceiro cenário simulado, cenário no qual foi simulado um teste com 60 itens e respondido por 10000 indivíduos.

A Tabela 19 apresenta o tempo médio, em horas, até se obter 10000 iterações e o tempo médio, em segundos, por iteração.

	Béguin and Glas (2001)	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Tempo médio até o burn-in	1:24:23.25	1:29:08.25	1:42:44.00	1:59:45.25
Tempo médio por iteração	0.5	0.52	0.62	0.71

Tabela 19: Tempo computacional, médio, por iteração e até se obter o burn-in por algoritmo para estudo simulado 3

A Figura 8 apresenta comparativos dois-a-dois das funções de autocorrelação para a cadeia da função log-densidade a *posteriori* de todos os algoritmos para primeira repetição,

onde é possível visualizar maiores valores de função de autocorrelação para o algoritmo Metropolis I. O fator de ineficiência são, 77.75 para o algoritmo de Béguin and Glas (2001), 66.57 para o algoritmo de Gonçalves et al. (2018), 76.86 para o algoritmo Metropolis I e 57.62 para o algoritmo Metropolis II.

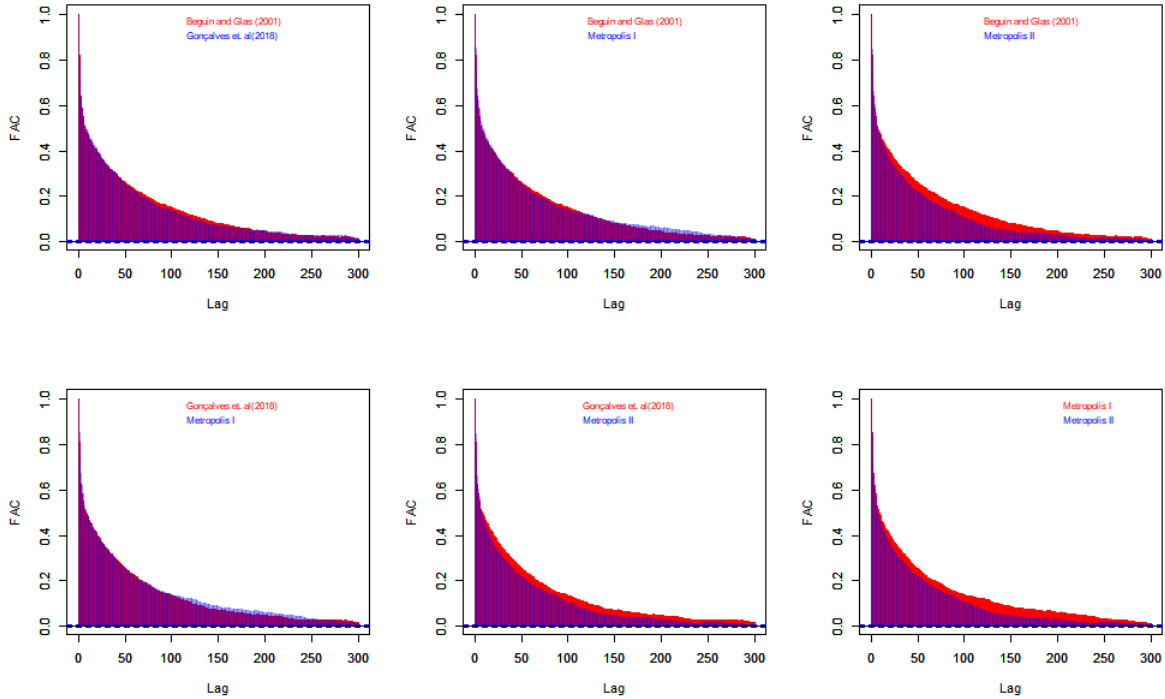


Figura 8: Comparação da função de autocorrelação da log-densidade a posteriori entre os quatro algoritmos no Estudo Simulado 3.

A Tabela 20 apresenta os valores de tamanho efetivo de amostra, para a função logaritmo da densidade a *posteriori*, para as 4 repetições dos 4 algoritmos após 10000 iterações, valor definido como burn-in, e de maneira que preservasse o denominador da equação (8), assim é possível detectar o impacto do fator de ineficiência.

	Béguin and Glas (2001)	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Repetição 1	1967	2022	2079	2733
Repetição 2	1837	2018	1983	2459
Repetição 3	1686	1958	1863	2305
Repetição 4	1929	2072	1860	2453
Média	1854	2017	1946	2487

Tabela 20: Tamanho Efetivo de Amostra após 10000 iterações para cada repetição no Estudo Simulado 3.

O algoritmo de Béguin and Glas (2001) e o algoritmo Metropolis I apresentam menores valores de tamanho efetivo de amostra. Algoritmo Metropolis II apresenta maiores valores de tamanho efetivo de amostra. Para os algoritmos de Béguin and Glas (2001), Metropolis I e Metropolis II a primeira repetição apresenta ótimos resultados, já o algoritmo de Gonçalves et al. (2018) a quarta repetição apresentou ótimos resultados.

A Tabela 21 apresenta a razão entre o tempo médio por iteração de cada algoritmo e a Tabela 22 apresenta a razão entre o valor médio do tamanho efetivo de amostra de cada uma das repetições, as leituras destas tabelas são similares as Tabela 5 e 6.

	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Béguin and Glas (2001)	1.05	1.25	1.41
Gonçalves et al. (2018)		1.19	1.35
Metropolis I			1.13

Tabela 21: Razão entre os tempos médios por iteração de cada algoritmo para Estudo Simulado 3.

	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Béguin and Glas (2001)	1.09	1.05	1.34
Gonçalves et al. (2018)		0.96	1.23
Metropolis I			1.28

Tabela 22: Razão entre o tamanho efetivo de amostra médio de cada algoritmo para Estudo Simulado 3.

Pode se perceber que o algoritmo Metropolis I apresenta baixa performance em relação aos demais algoritmos, mesmo possuindo um incremental ao resultado de Tamanho Efetivo de Amostra em relação ao algoritmo de Béguin and Glas (2001). O algoritmo Metropolis II apresenta melhores resultados de Tamanho Efetivo de Amostra, porém apresenta um custo computacional muito alto, apresentando baixa performance. Estes resultados são similares aos apresentados no estudo anterior, assim a ordem de eficácia em custo computacional para estes algoritmos é Gonçalves et al. (2018), Béguin and Glas (2001), Metropolis II e Metropolis I.

Porém estes resultados levam em consideração o valor médio. Devemos considerar, como nos estudos anteriores, a melhor performance. Para isto a Tabela 23 apresenta o tempo, em horas, necessário em cada algoritmo para se obter um tamanho efetivo de amostra de

1686. Quando consideramos a melhor performance temos a seguinte ordenação: Béguin and Glas (2001), Gonçalves et al. (2018), Metropolis II e Metropolis I.

	Béguin and Glas (2001)		Gonçalves et al. (2018)		Metropolis I		Metropolis II	
	Nº de iterações	Tempo	Nº de iterações	Tempo	Nº de iterações	Tempo	Nº de iterações	Tempo
Repetição 1	130708	18:04:02	128424	18:48:15	122563	21:08:25	96207	19:07:04
Repetição 2	139202	19:14:20	122394	17:45:54	125376	21:44:45	103429	20:17:29
Repetição 3	150000	20:41:07	117214	17:01:26	136279	23:31:09	108602	21:20:20
Repetição 4	121098	16:43:45	119651	17:21:03	138533	23:52:48	110194	21:37:57

Tabela 23: Número de iterações e tempo computacional necessário para se obter um tamanho efetivo de amostra fixo para Estudo Simulado 3

Para complementarmos os estudos, as Tabelas 24, 25 e 26 apresentam, respectivamente, a razão entre os tempos computacionais, a razão de tamanho efetivo de amostra da repetição, o número de iterações e tamanho efetivo de amostra para um tempo computacional fixo, das repetições de melhor desempenho de cada algoritmo.

	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Béguin and Glas (2001)	1.05	1.25	1.43
Gonçalves et al. (2018)		1.19	1.36
Metropolis I			1.14

Tabela 24: Razão entre os tempos por iteração das repetições de melhor desempenho de cada algoritmo para Estudo Simulado 3.

	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Béguin and Glas (2001)	1.02	1.08	1.41
Gonçalves et al. (2018)		1.06	1.4
Metropolis I			1.31

Tabela 25: Razão entre o tamanho efetivo de amostra das repetições de melhor desempenho de cada algoritmo para Estudo Simulado 3.

	Béguin and Glas (2001)	Gonçalves et al. (2018)	Metropolis I	Metropolis II
Total de iterações	150000	142488	119896	104293
Tamanho Efetivo de Amostra	1929	1887	1674	1859

Tabela 26: Número de iterações e tamanho efetivo de amostra, da repetição de melhor desempenho, com tempo computacional fixo para Estudo Simulado 3.

### 3.4 Comparação entre simulações

Nesta seção será apresentada a comparação de todos os algoritmos para os 3 estudos em relação ao tempo computacional e tamanho efetivo de amostra, considerando apenas a

repetição de melhor performance.

A Figura 9 mostra a evolução do tempo computacional, em segundos, por iterações demandado em cada um dos estudos simulados. Percebe-se que o tempo computacional demandado pelos algoritmos Metropolis cresce mais rápido em comparação aos algoritmos de Béguin and Glas (2001) e Gonçalves et al. (2018). Estes apresentam tempo computacionais próximos com pequenas diferenças entre eles. Quanto maior o número de itens e indivíduos maior será a diferença entre o tempo computacional demandado por cada algoritmo. A Figura 10 apresenta a razão entre o tempo computacional por iteração, de cada algoritmo, e o maior tempo computacional observado em cada estudo.

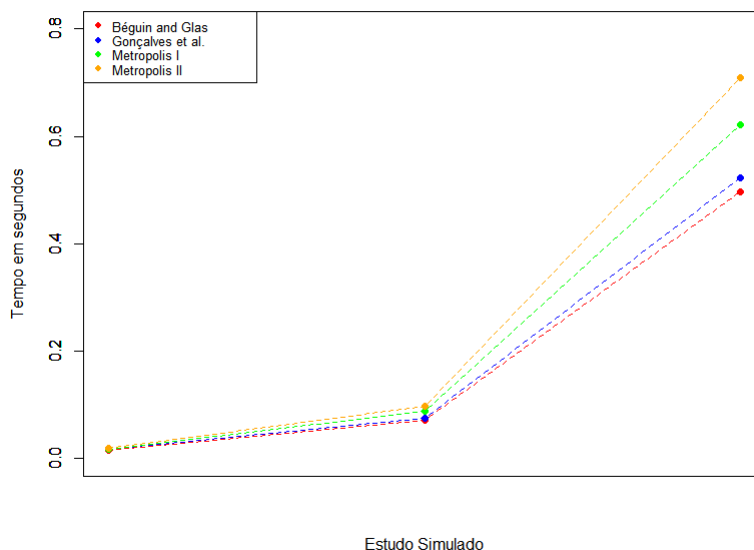
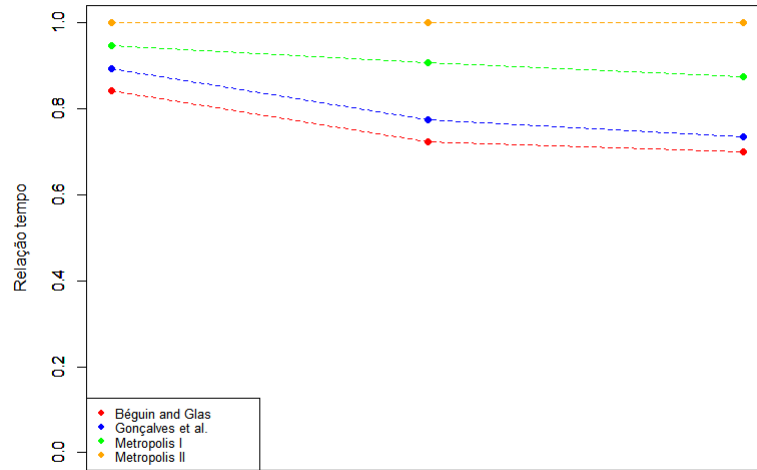


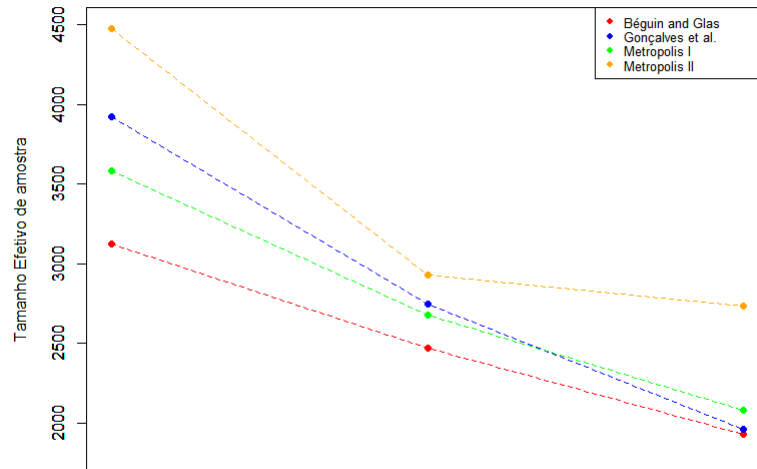
Figura 9: Comparação tempo por iteração das melhores performances de cada algoritmo.

A Figura 11 mostra o comportamento do tamanho efetivo de amostra no melhor desempenho de cada algoritmo para cada estudo. Para o cálculo de tamanho efetivo foi considerada uma amostra de 140000 iterações. É perceptível um comportamento comum entre os algoritmos, no qual o algoritmo Metropolis II sempre apresenta maiores tamanhos efetivos de amostra, seguido pelo algoritmo de Gonçalves et al. (2018), e por último de Béguin and Glas (2001). Algoritmo Metropolis I apresenta comportamento intermediário entre algoritmos de Gonçalves et al. (2018) e Béguin and Glas (2001), exceto pelo último estudo no qual seu valor foi superior ao de Gonçalves et al. (2018).



Estudo Simulado

Figura 10: Comparação razão entre o tempo por iteração de cada algoritmo e o máximo de tempo por iteração observado em cada estudo.



Estudo Simulado

Figura 11: Comparação tamanho efetivo de amostra das melhores performances de cada algoritmo.

A Figura 12 apresenta o tempo computacional, em segundos, necessário, pela performance de melhor desempenho de cada algoritmo, para se obter um tamanho efetivo de amostra fixo em cada estudo. Através dele há um grande indicio de que o algoritmo Metro-

polis I tem a pior performance, enquanto algoritmos de Béguin and Glas (2001) e Gonçalves et al. (2018) apresentam comportamentos bem similares. A Figura 13 apresenta razão entre o tempo computacional para se obter um tamanho de amostra fixo, da melhor performance de cada algoritmo, e o máximo entre estes por estudo.

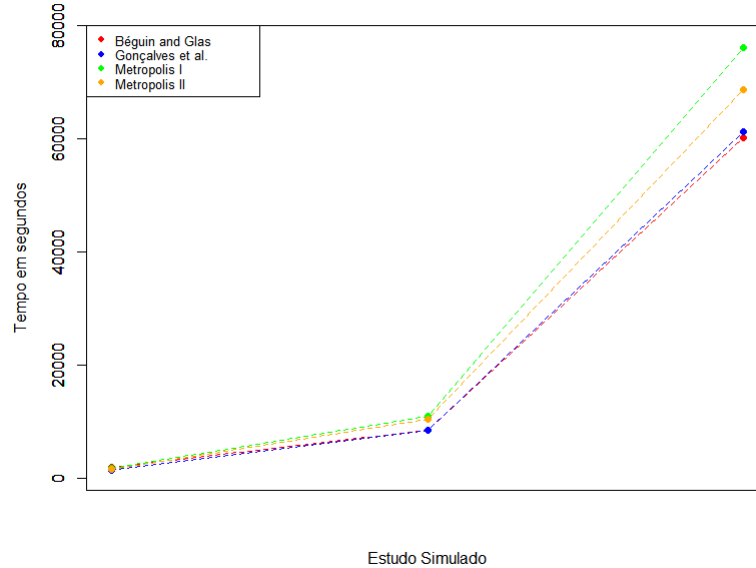
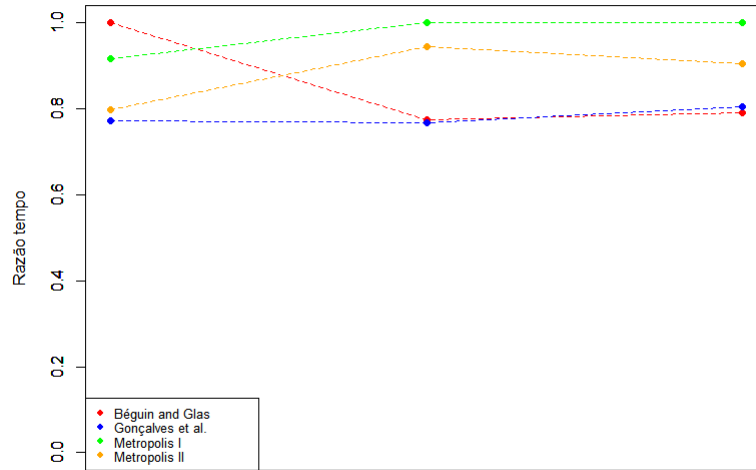


Figura 12: Comparação do tempo, em segundos, para a melhores performances de cada algoritmo obter um tamanho efetivo de amostra fixo.

As Figuras 14 e 15 apresentam o tamanho efetivo de amostra e o número de iterações obtidos quando se é fixado um tempo para cada algoritmo. Pode-se perceber que ao aumentar o número de itens e indivíduos o decaimento do tamanho efetivo de amostra para o algoritmo de Béguin and Glas (2001) é mais suave em comparação aos demais, os algoritmos Metropolis apresentam uma grande queda no número de iterações e tamanho efetivo de amostra do primeiro para o segundo estudo.

Cada algoritmo apresentou uma repetição diferente como melhor desempenho em cada estudo. Para o algoritmo de Béguin and Glas (2001) foram 3ª, 1ª e 4ª repetições, respectivamente. Para Gonçalves et al. (2018) 2ª, 2ª e 3ª, Metropolis I 2ª, 2ª e 1ª, Metropolis II 4ª, 3ª e 1ª.

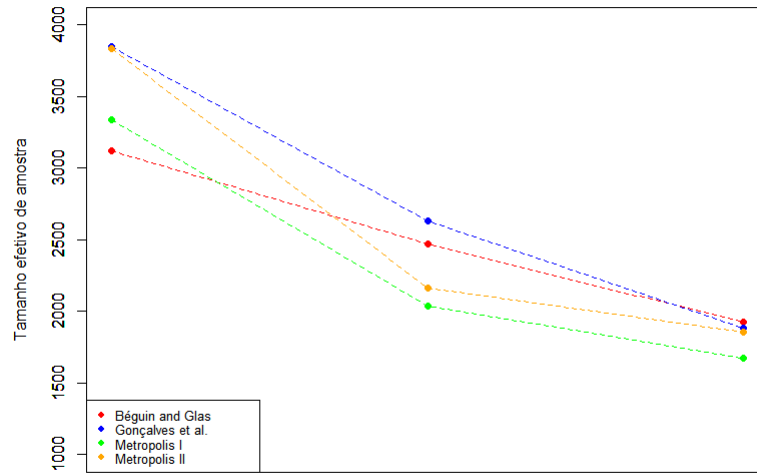




Estudo Simulado

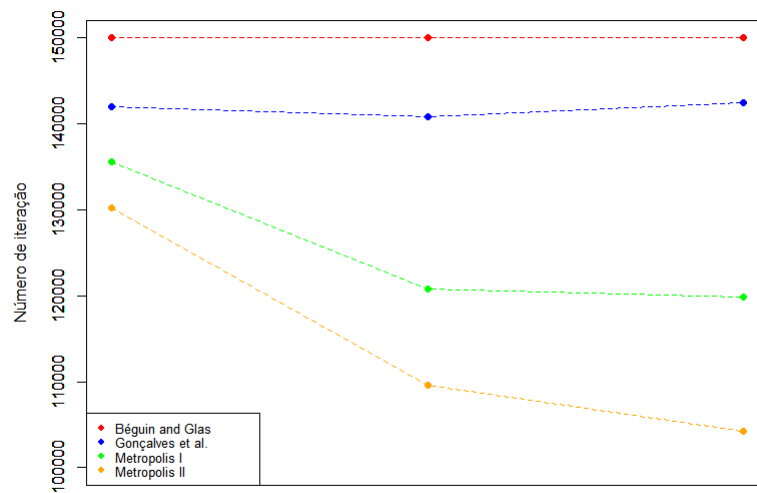
Figura 13: Comparação razão entre o tempo para se obter um tamanho efetivo de amostra fixo de cada algoritmo e o máximo de tempo para se obter um tamanho efetivo de amostra fixo de cada estudo.

Pode-se perceber que a ordenação por melhor desempenho, considerando o algoritmo que obtém um maior tamanho efetivo de amostra em um tempo fixo (assim é considerado a velocidade do algoritmo e os fatores de autocorrelação, ver Figura 14 e Figura 15), em cada estudo é a seguinte, para o primeiro estudo: Gonçalves et al. (2018), Metropolis II, Metropolis I, Béguin and Glas (2001); para segundo estudo: Gonçalves et al. (2018), Béguin and Glas (2001), Metropolis II e Metropolis I; para o terceiro estudo: Béguin and Glas (2001), Gonçalves et al. (2018), Metropolis II e Metropolis I. O algoritmo de Béguin and Glas (2001) apresenta uma melhora de performance conforme o número de itens e indivíduos aumenta, já o algoritmo de Gonçalves et al. (2018) apresenta uma boa performance em todos os estudos. Os algoritmos Metropolis, além de serem superados pelos outros dois algoritmos, têm sua melhor performance condicionada a uma calibração adequada das proposta para se obter se obter taxa de aceitação dentro dos padrões, esta calibração por si só não é trivial.



Estudo Simulado

Figura 14: Comparação do tamanho efetivo da amostra, das melhores performances de cada algoritmo, considerando tempo fixo.



Estudo Simulado

Figura 15: Comparação do número de iterações, das melhores performances de cada algoritmo, considerando tempo fixo.

## 4 Extensão do algoritmo de Gonçalves et al. (2018) para modelo 4 parâmetros

Culpepper (2016) apresenta uma extensão do algoritmo de Béguin and Glas (2001) para o modelo de Teoria de Resposta ao Item com 4 parâmetros. Nesta seção será apresentada uma extensão do algoritmo de Gonçalves et al. (2018) para o modelo de 4 parâmetros.

### 4.1 Algoritmo

O algoritmo proposto, assim como o algoritmo de Gonçalves et al. (2018), considera duas variáveis auxiliares para a construção de um algoritmo Gibbs-Sampling. O modelo de 4 parâmetros, Equação (1), pode ser reescrito utilizando duas variáveis auxiliares  $Z$  e  $X$  onde  $Z_{ij} \sim Multi(1, c_i, 1 - d_i - c_i, d_i)$ , com  $(X_{ij}|Z_{ij} = (0, 1, 0)) \sim N(a_i\theta_j - b_i, 1)$  e  $X_{ij}$  é um ponto de massa em zero, caso contrário. Assim, o modelo na Equação (1) pode ser reescrita da seguinte maneira.

$$Y_{ij} = \begin{cases} 1 & \text{se } Z_{ij} = (1, 0, 0) \text{ ou } (X_{ij} > 0, Z_{ij} = (0, 1, 0)) \\ 0 & \text{se } (X_{ij} < 0, Z_{ij} = (0, 1, 0)) \text{ ou } Z_{ij} = (0, 0, 1). \end{cases}$$

Pode-se mostrar que a Equação (1) é preservada quando se marginalizada com respeito a  $X$  e  $Z$ :

$$\begin{aligned} P(Y_{ij} = 1) &= P(Z_{ij} = (1, 0, 0)) + P(X_{ij} > 0|Z_{ij} = (0, 1, 0))P(Z_{ij} = (0, 1, 0)) \\ &= c_i + (1 - c_i - d_i) \int_0^\infty \frac{1}{\sqrt{(2\pi)}} e^{-\frac{1}{2}(X_{ij}-a_i\theta_j+b_i)^2} \partial X_{ij} \\ &= c_i + (1 - c_i - d_i)\Phi(a_i\theta_j - b_i). \end{aligned}$$

Quando  $d_i = 0 \forall i$  temos o modelo do algoritmo proposto por Gonçalves et al. (2018), pois  $Z_{ij} = Multi(1, c_i, 1 - c_i, 0)$  que pode ser reescrita como  $Binomial(c_i)$ .

Desta maneira é desejado obter amostra da distribuição a *posteriori* de  $\Psi = \{\mathbf{Z}, \mathbf{X}, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \theta\}$ , onde  $\mathbf{a} = (a_1, a_2, \dots, a_I)$ ,  $\mathbf{b} = (b_1, b_2, \dots, b_I)$ ,  $\mathbf{c} = (c_1, c_2, \dots, c_I)$ ,  $\mathbf{d} = (d_1, d_2, \dots, d_I)$ ,  $\theta = (\theta_1, \theta_2, \dots, \theta_J)$  e  $X$  e  $Z$  são matrizes de tamanho  $I \times J$  com entradas  $X_{ij}$  e  $Z_{ij}$  com  $i=1 \dots I$

e  $j=1\dots J$ .

Se consideramos os seguintes blocos  $\Psi_1 = \{\mathbf{Z}, \mathbf{X}\}$ ,  $\Psi_2 = \{\theta\}$ ,  $\Psi_3 = \{\mathbf{a}, \mathbf{b}\}$  e  $\Psi_4 = \{\mathbf{c}, \mathbf{d}\}$ , é possível amostrar diretamente das respectivas distribuições condicionais completas. Utilizando a premissa de independência condicional entre itens e indivíduos, as distribuições condicionais completas podem ser escritas das seguintes maneiras:

$$\pi(\Psi_1|\cdot) = \prod_{i=1}^I \prod_{j=1}^J \pi(X_{ij}, Z_{ij}|\cdot),$$

$$\pi(\Psi_2|\cdot) = \prod_{j=1}^J \pi(\theta_j|\cdot),$$

$$\pi(\Psi_3|\cdot) = \prod_{i=1}^I \pi(a_i, b_i|\cdot),$$

$$\pi(\Psi_4|\cdot) = \prod_{i=1}^I \pi(c_i, d_i|\cdot).$$

Para  $\Psi_1$ , tem-se que:

$$\pi(Z_{ij}, X_{ij}|\cdot) \propto \pi(Y_{ij}|Z_{ij}, X_{ij})\pi(Z_{ij}|c_i, d_i)\pi(X_{ij}|a_i, b_i, \theta_j).$$

Após alguns cálculos apresentados no Apêndice B para  $Y_{ij} = 1$ , tem-se que  $Z_{ij} = (1, 0, 0)$  se  $w_{ij} = 0$ , e  $Z_{ij} = (0, 1, 0)$  se  $w_{ij} = 1$ , onde  $w_{ij} \sim \text{Bernoulli}\left(\frac{(1-d_i-c_i)\Phi(a_i\theta_j-b_i)}{c_i+(1-d_i-c_i)\Phi(a_i\theta_j-b_i)}\right)$  e  $X_{ij}$  é um ponto de massa em zero se  $Z_{ij} = (1, 0, 0)$  e  $X_{ij} \sim N_{(0,\infty)}(a_i\theta_j-b_i, 1)$  se  $Z_{ij} = (0, 1, 0)$ .

Para  $Y_{ij} = 0$ , tem-se que  $Z_{ij} = (0, 0, 1)$  se  $v_{ij} = 1$  e  $Z_{ij} = (0, 1, 0)$  se  $v_{ij} = 0$ , onde  $v_{ij} \sim \text{Bernoulli}\left(\frac{d_i}{d_i+(1-c_i-d_i)(1-\Phi(a_i\theta_j-b_i))}\right)$ , e  $X_{ij}$  é um ponto de massa em zero se  $Z_{ij} = (0, 0, 1)$  e  $X_{ij} = N_{(-\infty,0)}(a_i\theta_j-b_i, 1)$  se  $Z_{ij} = (0, 1, 0)$ .

Para  $\Psi_2$ , utilizando como *priori*  $\text{Normal}(0,1)$ , tem-se que:

$$\pi(\theta_j|\cdot) \propto \pi(X_{ij}|a_i, b_i, \theta_j)\pi(\theta_j)$$

Após alguns cálculos (ver Apêndice B):

$$(\theta_j | \cdot) \sim N \left( \frac{\sum_{i; Z_{ij}=(0,1,0)} a_i (X_{ij} - b_i)}{\sum_{i; Z_{ij}=(0,1,0)} a_i^2 + 1}, \frac{1}{\sum_{i; Z_{ij}=(0,1,0)} a_i^2 + 1} \right).$$

Para  $\Psi_3$  são atribuídas as seguintes distribuições a priori para os parâmetros  $a_i, b_i, a_i \sim N_{[0, \infty)}(\mu_a, \sigma_a^2), b_i \sim N(\mu_b, \sigma_b^2)$ , tem-se que:

$$\pi(a_i, b_i | \cdot) \propto \prod_{j=1}^J \pi(X_{ij} | a_i, b_i, \theta_j) \pi(a_i) \pi(b_i),$$

$$(a_i, b_i | \cdot) \sim N_2(\mu, \Sigma) I_{a_i > 0},$$

$$\mu = \begin{bmatrix} \mu_a^* \\ \mu_b^* \end{bmatrix}, \Sigma = \begin{bmatrix} \sigma_a^{2*} & \sigma_a^* \sigma_b^* \gamma \\ \sigma_a^* \sigma_b^* \gamma & \sigma_b^{2*} \end{bmatrix}, \text{ onde:}$$

$$\sigma_a^{2*} = \frac{\sigma_a^2}{(\sigma_a^2 \sum_{j; Z_{ij}=(0,1,0)} \theta_j^2 + 1)(1 - \gamma^2)},$$

$$\sigma_b^{2*} = \frac{\sigma_b^2}{(\sigma_b^2 N_J + 1)(1 - \gamma^2)},$$

onde  $N_J = \sum_{j=1}^J I_{\{Z_{ij}=(0,1,0)\}}$ ,

$$\gamma = \frac{\sigma_a \sigma_b \sum_{j; Z_{ij}=(0,1,0)} \theta_j}{[(\sigma_a^2 \sum_{j; Z_{ij}=(0,1,0)} \theta_j^2 + 1)(\sigma_b^2 N_J + 1)]^{1/2}},$$

$$\mu_a^* = \sigma_a^{2*} \left( \sum_{j; Z_{ij}=(0,1,0)} X_{ij} \theta_j + \mu_a \sigma_a^{-2} \right) - \sigma_a^* \sigma_b^* \gamma \left( \sum_{j; Z_{ij}=(0,1,0)} X_{ij} - \mu_b \sigma_b^{-2} \right),$$

$$\mu_b^* = \sigma_a^* \sigma_b^* \gamma \left( \sum_{j; Z_{ij}=(0,1,0)} X_{ij} \theta_j + \mu_a \sigma_a^{-2} \right) - \sigma_b^{2*} \left( \sum_{j; Z_{ij}=(0,1,0)} X_{ij} - \mu_b \sigma_b^{-2} \right).$$

Para  $\Psi_4$ , adotamos uma distribuição a priori  $(c_i, 1 - d_i - c_i, d_i) \sim Dir(\alpha_1, \alpha_2, \alpha_3)$ ,

tem-se que:

$$\pi(c_i, 1 - c_i - d_i, d_i | \cdot) \propto \pi(Z_{ij} | c_i, d_i) \pi(c_i, d_i),$$

$$((c_i, 1 - c_i - d_i, d_i) | \cdot) \sim Dir \left( \alpha_1 + \sum_{j=1}^J Z_{ij1}, \alpha_2 + \sum_{j=1}^J Z_{ij2}, \alpha_3 + \sum_{j=1}^J Z_{ij3} \right).$$

A estrutura do algoritmo é apresentada no Algoritmo 4.

---

**Algoritmo 4:** Modelo de 4 parâmetros

---

**Entrada:** Banco de dados, valor inicial para os parâmetros, valores para os hiperparâmetros (parâmetro das distribuições *a priori*), número de iterações desejadas.

**Saída:** Amostras de cada parâmetro e traço latente.

```

1 início
2   para  $k$  de 1 até o número de iterações faça
3     Gere uma amostra de  $\Psi_1$ , utilizando a distribuição condicional completa.
4      $(X_{ij}, Z_{ij} | \cdot)$ ;
5     Gere uma amostra de  $\Psi_2$ , através da distribuição de  $(\theta_j | \cdot)$ ;
6     Gere uma amostra de  $\Psi_3$ , através da distribuição de  $(a_i, b_i | \cdot)$ ;
7     Gere uma amostra de  $\Psi_4$ , através da distribuição de  $(c_i, 1 - c_i - d_i, d_i | \cdot)$ ;
8   fim
9 fim

```

---

## 4.2 Estudo simulado

Nesta seção serão apresentados resultados de dois estudos simulados para o algoritmo de 4 parâmetros. O primeiro estudo foi realizado em uma amostra simulada de 2000 indivíduos e 40 itens. O segundo, para uma amostra de 10000 indivíduos e 60 itens. Os parâmetros de discriminação foram gerados através de uma Uniforme(0.8,2.2), os parâmetros de dificuldade através de Uniforme(-3,3), os parâmetros de acerto casual através de Uniforme(0,0.15) e para assintota superior Uniforme(0,0.05).

### 4.2.1 Estudo simulado 1

Nesta seção serão apresentadas as análises realizadas para o primeiro caso simulado, amostra de 2000 indivíduos e 40 itens. Foram realizadas 150000 iterações e descartadas as 10000 primeiras.

A Figura 16 apresenta gráficos das estimativas *a posteriori* (média) de cada parâmetro dos itens versus valores reais utilizados para a simulação. Quanto mais perto os pontos estão da reta que corta a figura mais próxima foram as estimativas dos valores reais. E a Figura 17 apresenta a distribuição real e estimada dos traços latentes.

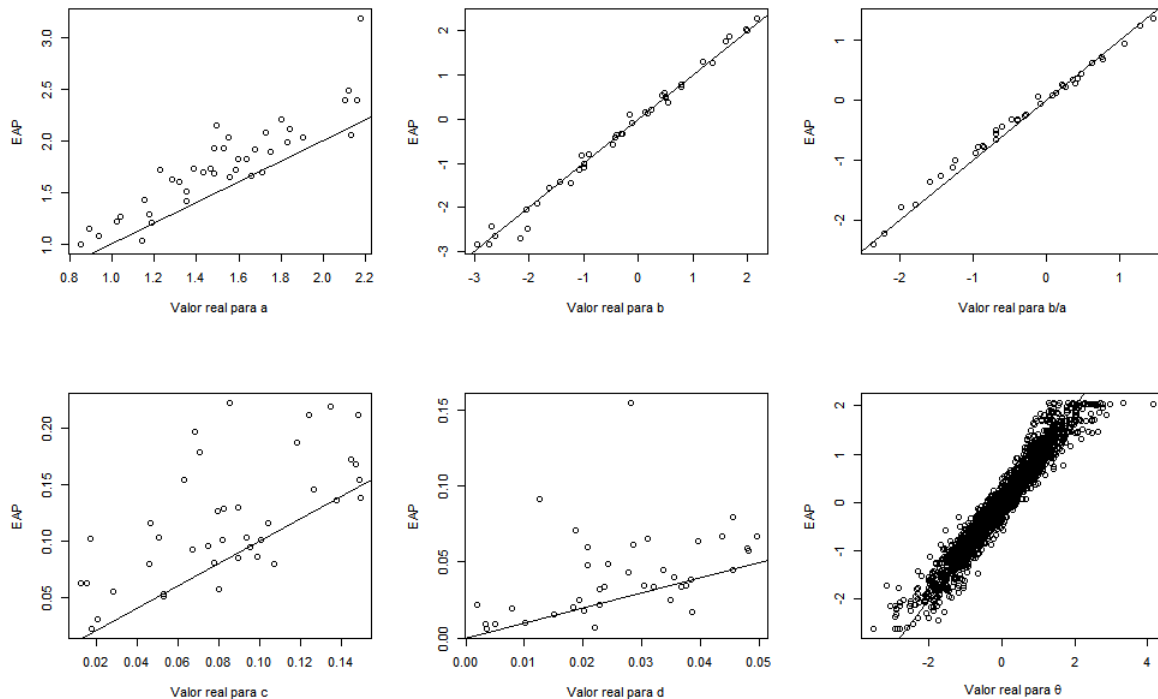


Figura 16: Estimativas dos parâmetros na primeiro estudo simulado.

Percebe-se que o parâmetro de dificuldade é o parâmetro que há estimativas mais próximas do seu valor real. O parâmetro de discriminação apresenta estimativas maiores que os valores reais. O parâmetro de assíntota superior apresenta algumas estimativas bem destoantes dos valores reais para dois itens. Percebe-se que para indivíduos com traço latentes altos, acima de 2, seus valores são subestimados, isto se dá pois não há itens com parâmetro de dificuldade superior a 2.

A Figura 18 apresenta a CCI para o 29º item do estudo simulado, demonstrando o comportamento geral das CCI para este estudo. Apresentando comparação entre a curva estimada e a curva real do modelo.

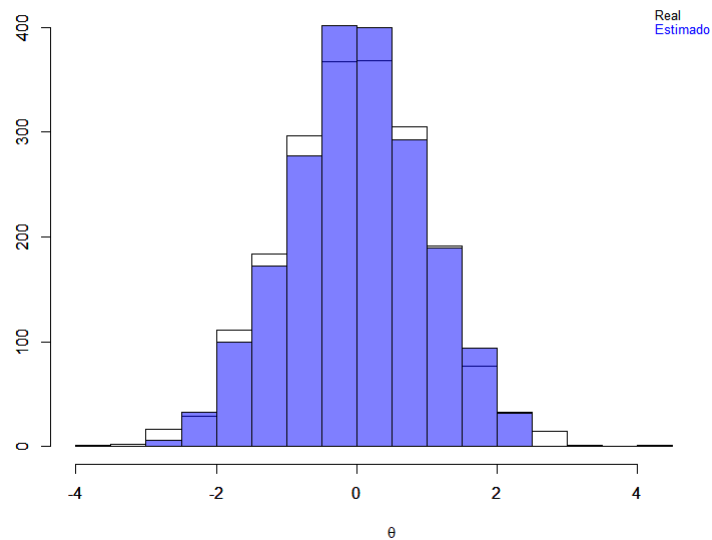


Figura 17: Distribuição do valor real e estimativa dos traços latentes para primeiro estudo simulado.

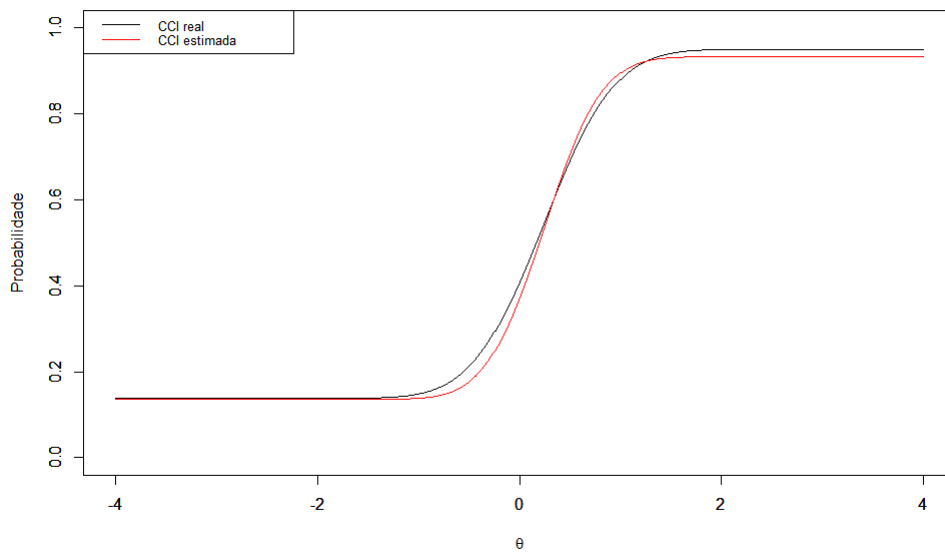


Figura 18: CCI 29º item. Valor real:  $a=1.8$ ;  $b=0.43$ ;  $c=0.14$ ,  $d=0.05$ . Estimativa:  $a=2.2$ ;  $b=0.53$ ;  $c=0.14$ ;  $d=0.07$ .

#### 4.2.2 Estudo simulado 2

Nesta seção serão apresentadas as análises realizadas para o primeiro caso simulado, amostra de 10000 indivíduos e 60 itens. Foram realizadas 150000 iterações e descartadas as 10000



primeiras.

A Figura 19 apresenta gráficos das estimativas *a posteriori* (média) de cada parâmetro dos itens versus valores reais utilizados para a simulação e a Figura 20 apresenta a distribuição real e estimada dos traços latentes.

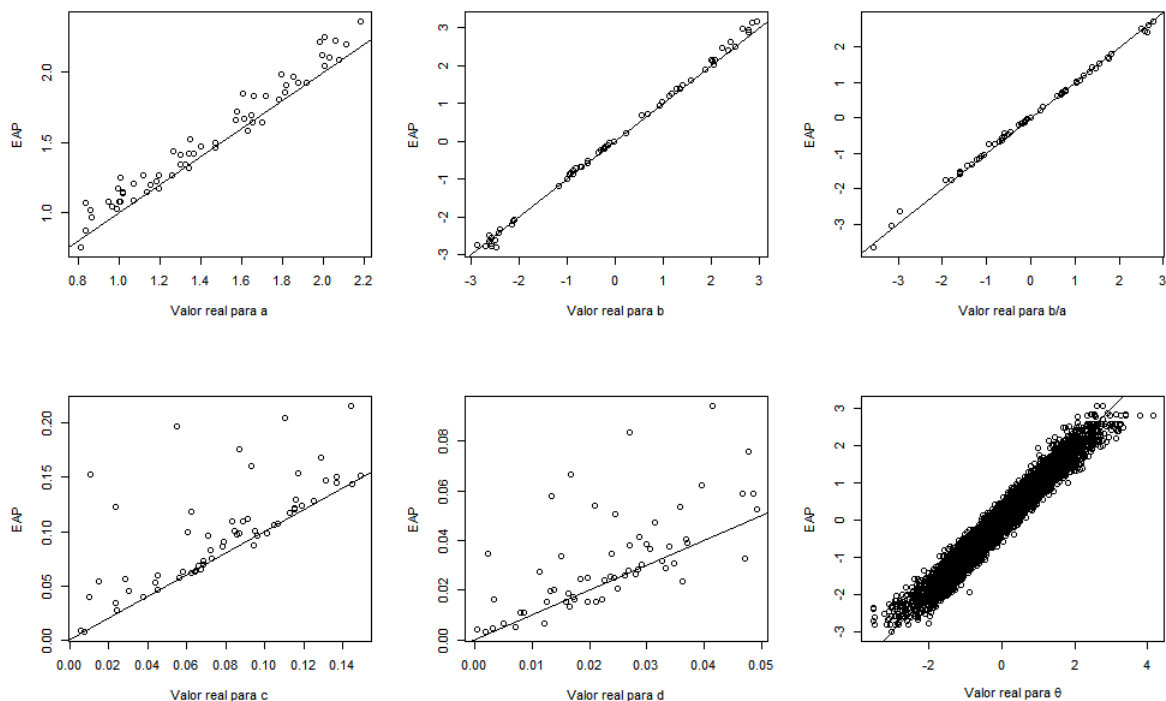


Figura 19: Estimativas dos parâmetros para segundo estudo simulado.

Percebe-se que o parâmetro de dificuldade é o parâmetro que há estimativas mais próximas do seu valor real. O parâmetro de discriminação apresenta estimativas maiores que os valores reais, porém próximas aos valores reais, em comparação aos resultados obtidos para a primeira simulação Figura (16). O parâmetro de assintota superior apresenta estimativas mais próximos dos reais e valores menos dispersos. O parâmetro de acerto casual apresenta estimativas superiores aos valores reais, porém mais próximas aos valores reais e menos dispersas.

Percebe-se que para indivíduos com traço latentes altos, acima de 2, as suas estimativas são menos subestimadas em relação aos resultados obtidos para a primeira simulação (Figura 16).

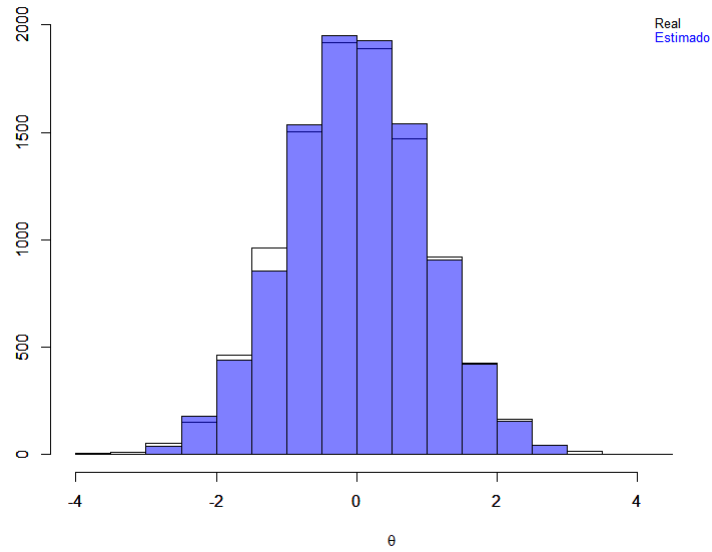


Figura 20: Distribuição do valor real e estimativa dos traços latentes para segundo estudo simulado.

A Figura 21 apresenta a CCI para o 50º item do estudo simulado, mostrando as diferenças entre a curva estimada e a curva real.

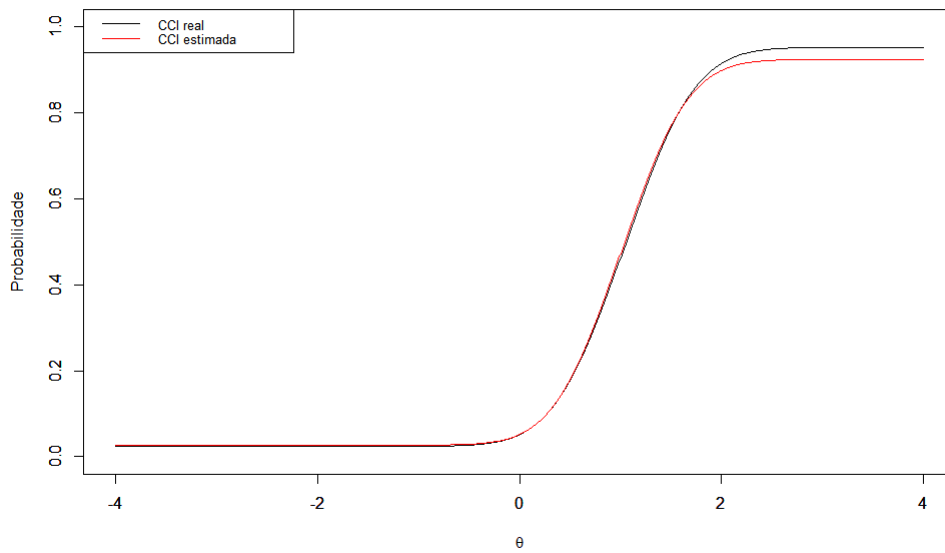


Figura 21: CCI 50º item. Valor real:  $a=1.82$ ;  $b=1.89$ ;  $c=0.02$ ,  $d=0.05$ . Estimativa:  $a=1.9$ ;  $b=1.91$ ;  $c=0.03$ ;  $d=0.08$

### 4.3 Comparação entre os modelos de 3 e 4 parâmetros

Nesta seção será apresentado um estudo comparativo entre as estimativas para o modelo de 3 parâmetros utilizando o algoritmo proposto na Seção 4.1 e as estimativas para o modelo de 4 parâmetros utilizando o algoritmo de Gonçalves et al. (2018).

Inicialmente foi ajustado o algoritmo de 4 parâmetros para o conjunto de dados gerado do modelo de 3 parâmetros apresentado na Seção 3.3. Foram realizadas 150000 iterações e utilizado um valor de 10000 iterações como burn-in. A Figura 22 apresenta as estimativas a *posteriori*, média, para os parâmetros  $a$ ,  $b$ ,  $c$  e para  $b/a$  (dificuldade do item, ver equação (1) e equação (2)) e as Figuras 23 e 24 apresentam estimativas a *posteriori* para os valores de  $\theta$ .

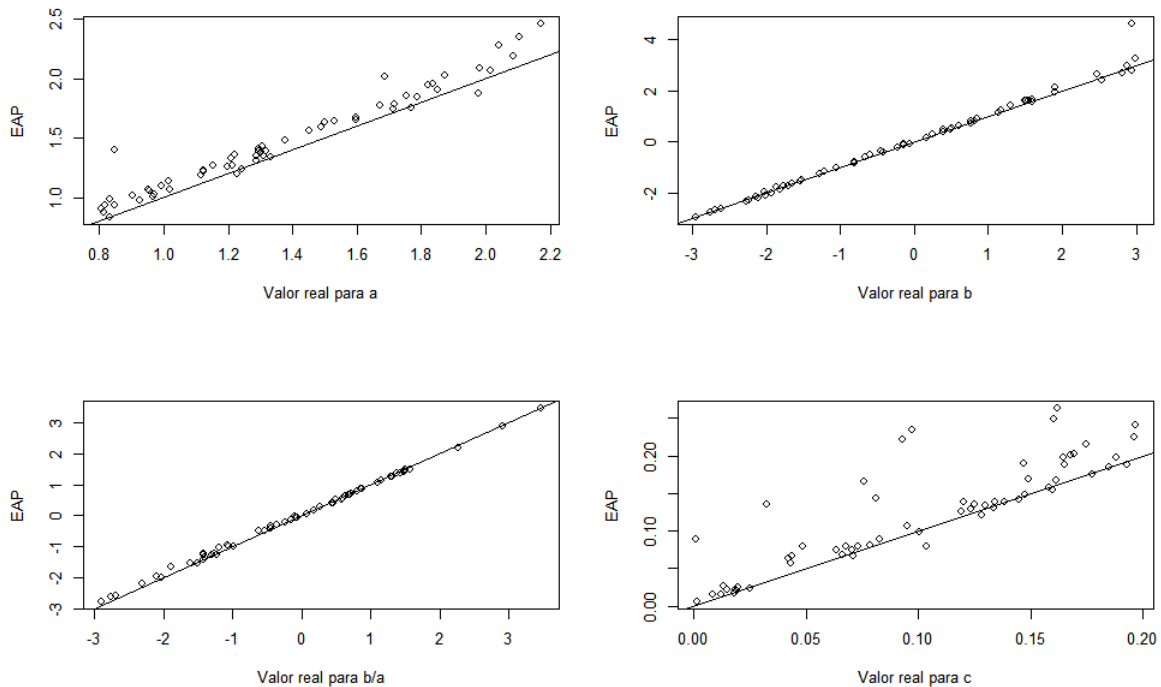


Figura 22: Estimativas para parâmetros dos itens ajustando o modelo 4PNO aos dados gerados pelo 3PNO.

A Figura 25 apresenta os valores de estimativa para o parâmetro  $d$ , é notável que este apresenta valores baixos, valor máximo próximo a 0.05, o que sugere que o modelo 4PNO ajusta bem dados gerados no modelo 3PNO.

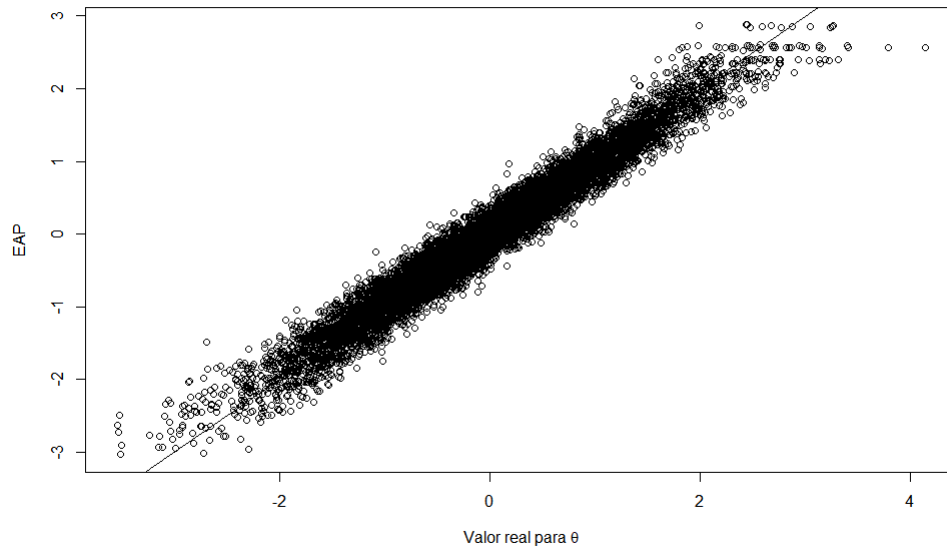


Figura 23: Estimativas para o traço latente ajustando o modelo 4PNO aos dados gerados pelo 3PNO.

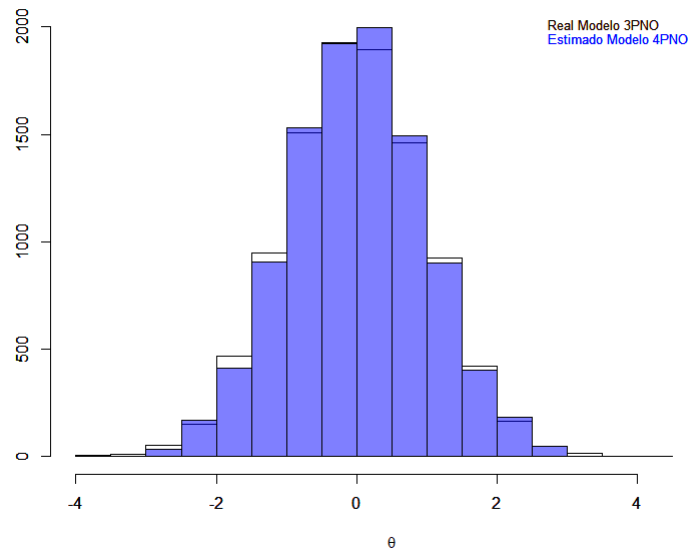


Figura 24: Distribuição das estimativas para o traço latente ajustando o modelo 4PNO aos dados gerados pelo modelo 3PNO.

A Figura 26 apresenta a Curva Característica do 41º item do banco simulado, representando o comportamento geral das CCI, percebe-se que a curva estimada no modelo de 4 parâmetros é próxima a curva real. Logo o modelo de 4 parâmetros é capaz de se ajustar

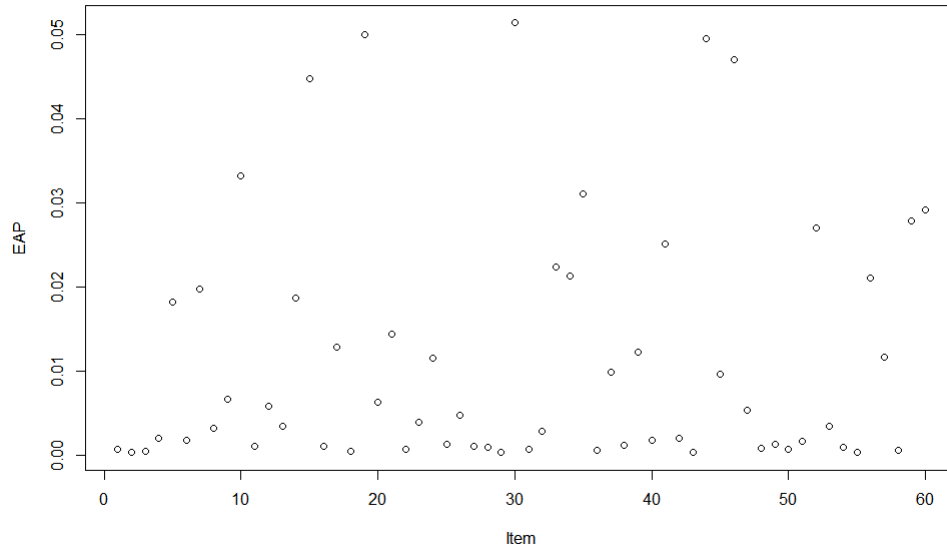


Figura 25: Estimativas para assintota superior ajustando o modelo 4PNO aos dados gerados pelo 3PNO.

bem a dados gerados pelo modelo de 3 parâmetros.

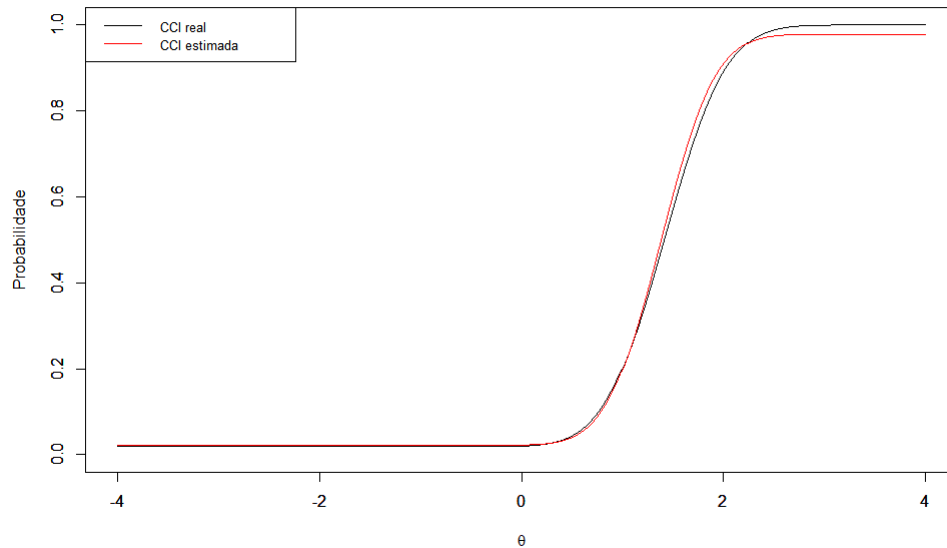


Figura 26: CCI 41º item. Valor real:  $a=2.11$ ;  $b=3$ ;  $c=0.02$ . Estimativa:  $a=2.35$ ;  $b=3.25$ ;  $c=0.02$ ;  $d=0.03$

Foi-se aplicado ao algoritmo de três parâmetros proposto por Gonçalves et al. (2018) ao

banco de dados simulado utilizando o modelo de 4PNO apresentado na Seção 4.2.2. A Figura 27 apresenta as estimativas *a posteriori* para os parâmetros  $a$ ,  $b$ ,  $c$  e para  $b/a$ . Percebe-se que o parâmetro  $b$  é estimado com valores superiores aos reais e os parâmetros  $a$  e  $c$  são mal estimados.

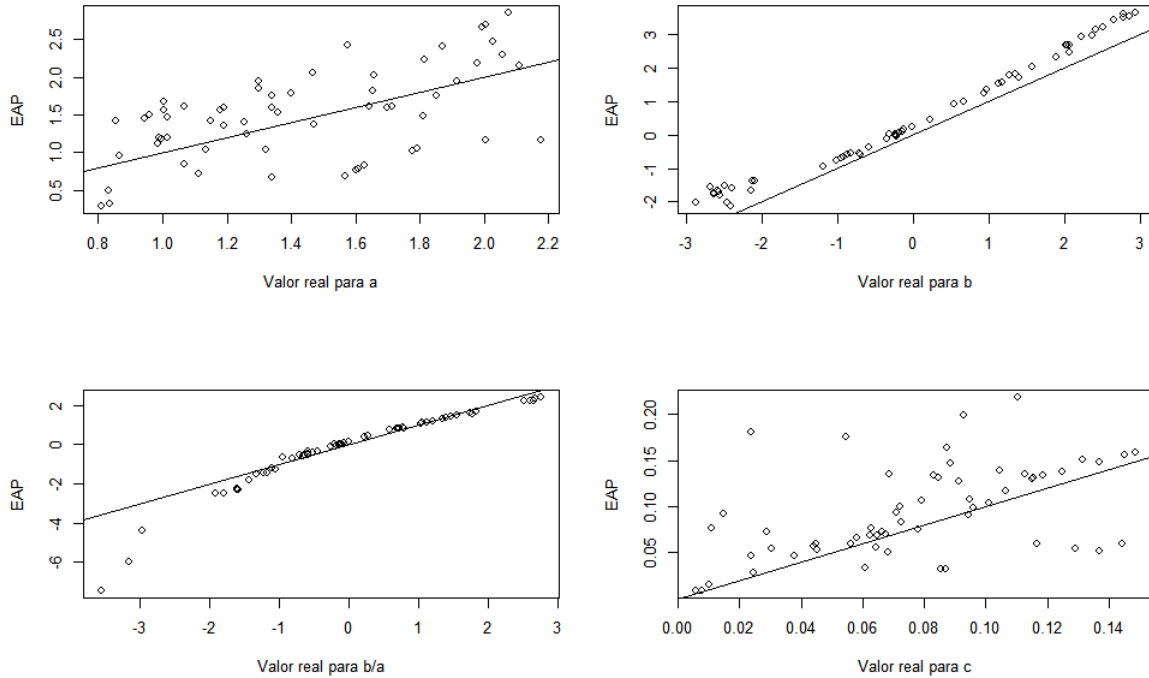


Figura 27: Estimativas para parâmetros dos itens ajustando o modelo 3PNO aos dados gerados pelo 4PNO.

As Figuras 28 e 29 apresentam as estimativas *a posteriori* para os valores de  $\theta$ . Para os traços latentes é observado que os valores inferiores e superiores são estimados abaixo do seu valor real, isto se dá ao fato de que itens com dificuldade baixa tem o valor deste parâmetro estimados com valores inferiores e a ausência do parâmetro  $d$ .

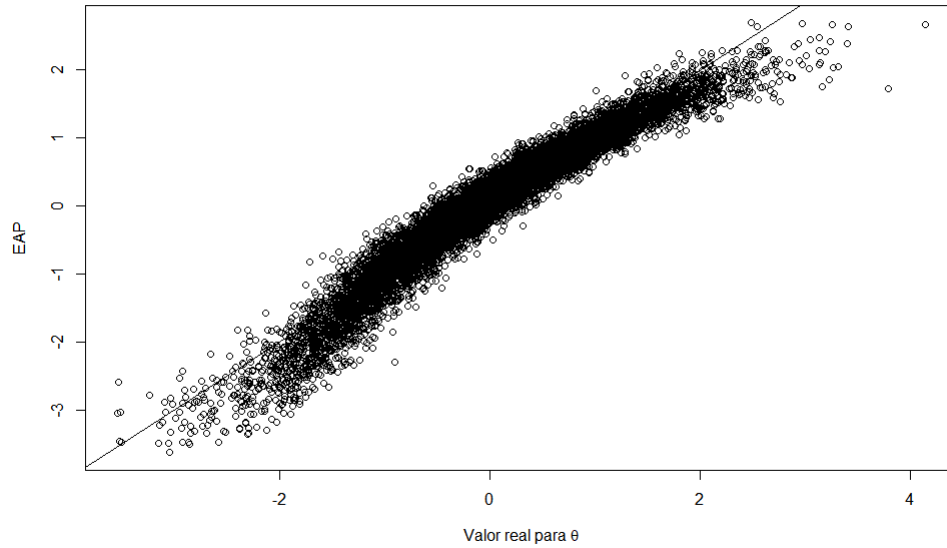


Figura 28: Estimativas para o traço latente ajustando o modelo 3PNO aos dados gerados pelo 4PNO.

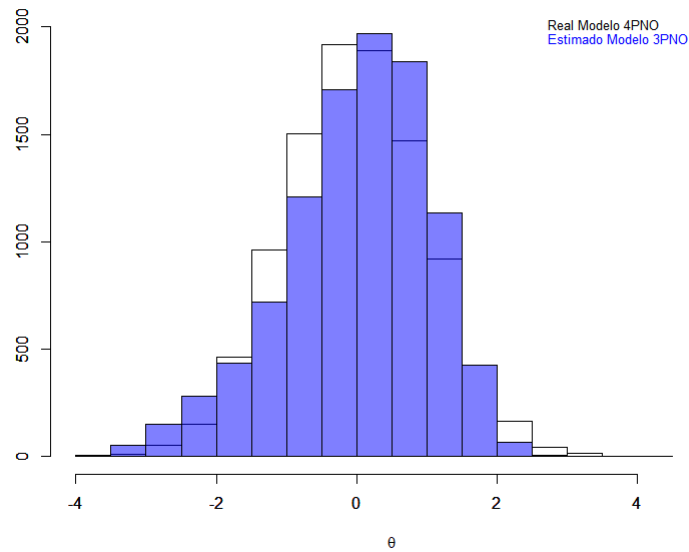


Figura 29: Distribuição das estimativas para o traço latente ajustando o modelo 3PNO aos dados gerados pelo modelo 4PNO.

A Figura 30 apresenta as CCI estimada e real para 13º item e a Figura 31 apresenta CCI estimada e real para 30º item (no qual o parâmetro  $d = 0.05$ ), estas figuras ilustram a possível causa de estimativas inferiores para  $\theta$ . Logo o algoritmo proposto por Gonçalves

et al. (2018) não é capaz de fazer boas estimativas para o modelo 4PNO.

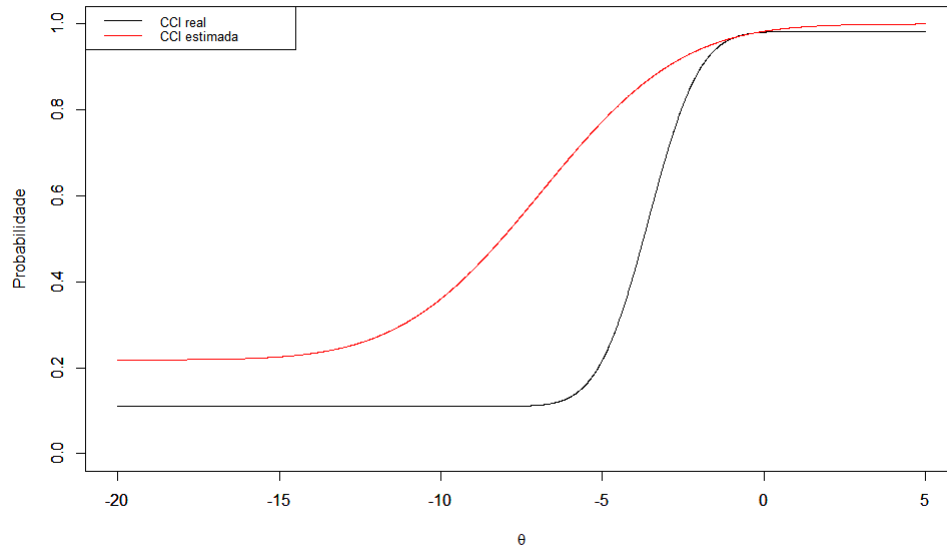


Figura 30: CCI 13º item. Valor real:  $a=0.81$ ;  $b=-2.88$ ;  $c=0.11$ ;  $d=0.02$ . Estimativa:  $a=0.29$ ;  $b=-2.01$ ;  $c=0.21$

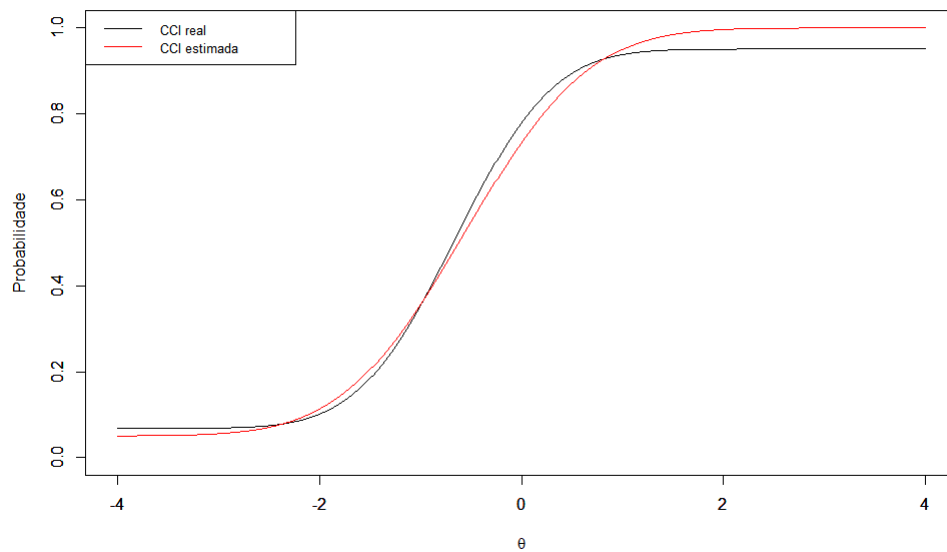


Figura 31: CCI 30º item. Valor real:  $a=1.32$ ;  $b=-0.86$ ;  $c=0.07$ ;  $d=0.05$ . Estimativa:  $a=1.04$ ;  $b=-0.58$ ;  $c=0.05$



## 5 Aplicação dos algoritmos ao banco de dados do Enem

Nesta seção os algoritmos utilizados para estimativas do modelo de 3 parâmetros e de 4 parâmetros de TRI serão aplicados a uma amostra de conjunto de dados do Exame Nacional do Ensino Médio (Enem) do Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) de 2017.

O ENEM possui o intuito de testar o conhecimento de alunos de Ensino Médio em 4 áreas Linguagens e códigos, Ciências Humanas, Ciências da Natureza, Matemática. Para esta aplicação foi selecionada uma amostra de 10000 indivíduos do banco de dados relacionado a proficiência de Matemática. Foram aplicados o algoritmo de Gonçalves et al. (2018) e o algoritmo construído e apresentado na Seção 4.

Para o algoritmo de 3 parâmetros foi utilizado um truncamento para que o parâmetro  $c$  não obtivesse valores superiores a 0.2, para o algoritmo de 4 parâmetros foi utilizado o mesmo truncamento para o parâmetro  $c$  e para o parâmetro  $d$ . Este truncamento se faz necessário, pois não são esperados itens com comportamentos diferentes a estes, valores de  $c$  ou  $d$  superiores poderia comprometer a estimação dos demais parâmetros do modelo.

Para o algoritmo de 3 parâmetros utilizou-se a metodologia de CDF inversa para se amostrar da distribuição Beta Truncada, a amostragem do valor de  $c$  foi realizada após 5000 iterações, para evitar problemas computacionais. Para o algoritmo de 4 parâmetros foi utilizado o método de amostragem condicional da distribuição Dirichlet (ver capítulo 7 de Ng et al. (2011)). Em ambos os casos foi selecionado um valor de burn-in de 20000 iterações.

A Figura 32 apresenta as estimativas *a posteriori* de ambos os modelos para o parâmetro  $a$ ,  $b$ ,  $c$ ,  $\theta$ . Pode-se perceber que o modelo de 4 parâmetros possui estimativas superiores para o parâmetro  $a$  em relação ao modelo de 3PNO. Nos demais parâmetros apenas alguns itens são estimados com valores diferentes em cada um dos modelos. A Figura 33 apresenta a distribuição da estimativa dos traço latentes de ambos os modelos, onde se é possível ver pequenas diferenças nas estimativas.

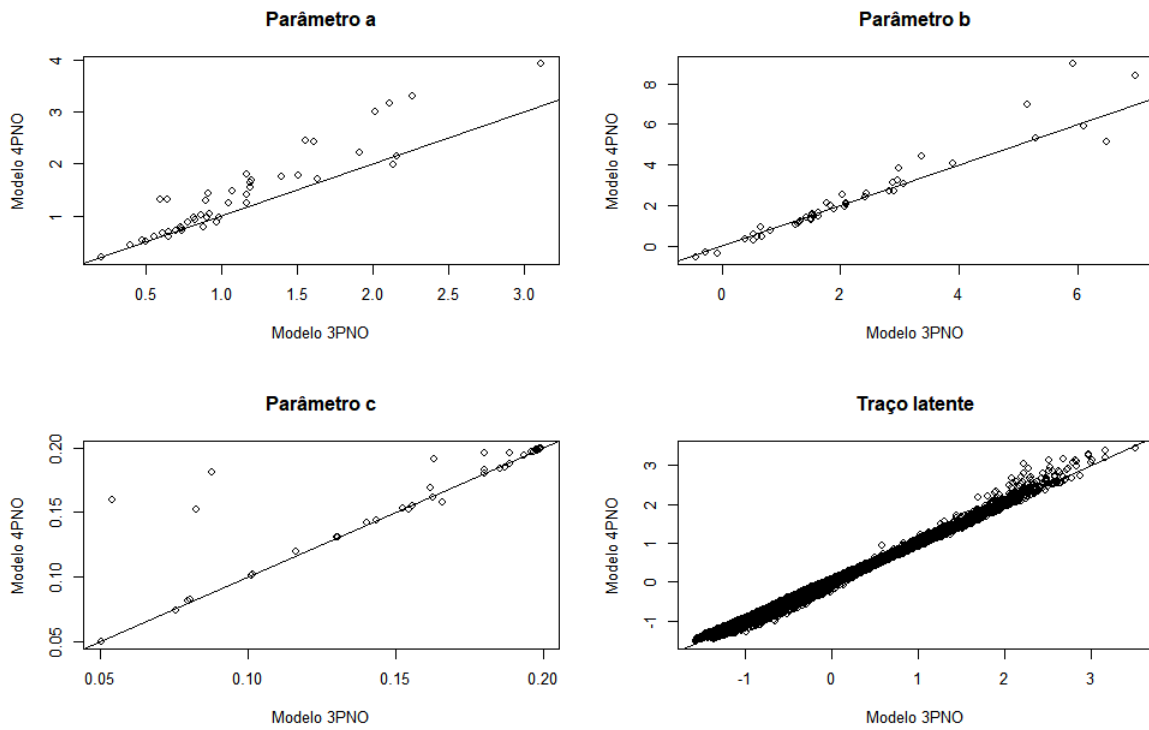


Figura 32: Estimativas para os parâmetros  $a$ ,  $b$ ,  $c$  e  $\theta$  utilizando os modelos de 3 parâmetros e 4 parâmetros ao banco de dados Enem.

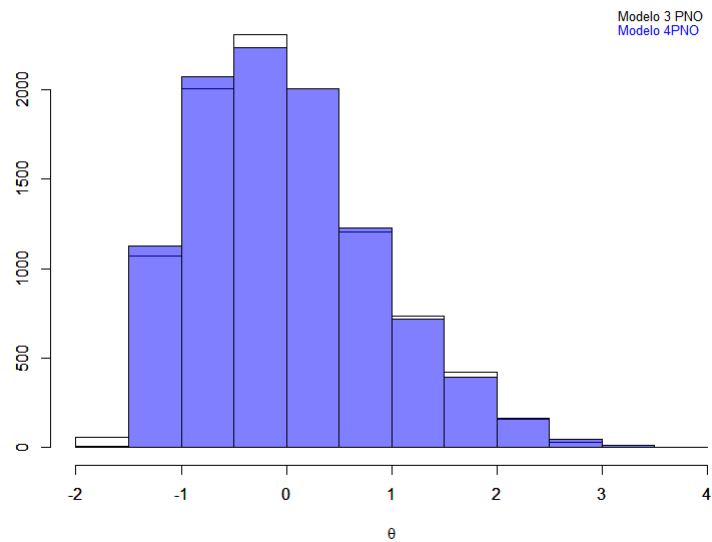


Figura 33: Distribuição das estimativas para o traço latente ajustando o modelo 3PNO e modelo 4PNO.

A Figura 34 apresenta as estimativas para o parâmetro  $d$ , percebe-se que vários valores foram superiores a 0.1 dando indícios de influência deste parâmetro nos itens, sugerindo a necessidade de um modelo mais complexo. A Figura 35 apresenta a CCI do item 19 representando as diferenças de estimativas de CCI de cada modelo.

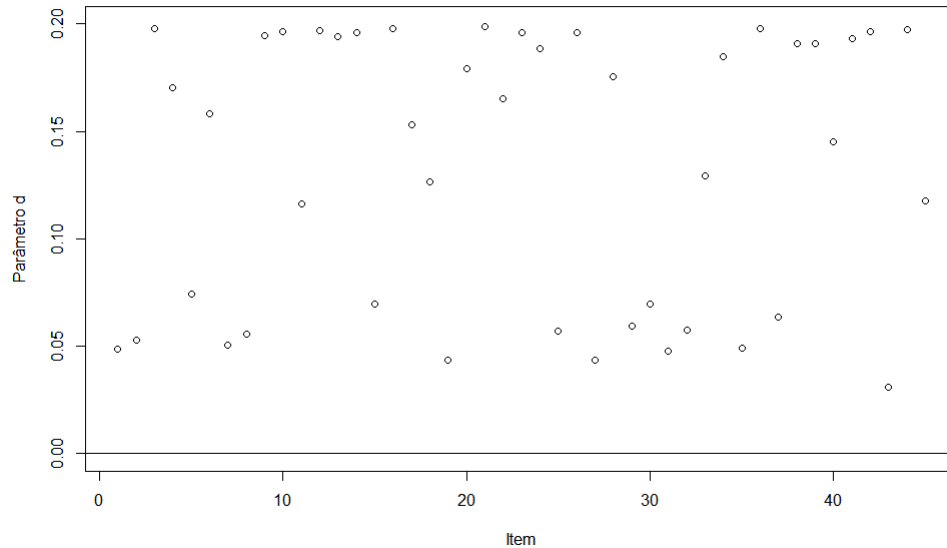


Figura 34: Estimativa para o parâmetro  $d$  no banco de dados Enem.

Em aspectos práticos é desejado ver se há diferença significativa na colocação de candidatos do ENEM em relação a aplicação do modelo de 3 parâmetros e 4 parâmetros. Figura 36 mostra a diferença, para um mesmo individuo, entre a colocação ao utilizar o modelo de 3 parâmetros e ao utilizar o modelo de 4 parâmetros, Tabela 27 apresenta as maiores diferenças entre a colocação dos dois modelos. Apenas 27 indivíduos obtiveram a mesma colocação em ambos os modelos, 5437 foram observados uma colocação maior no modelo de 3 parâmetros que o modelo de 4 parâmetros.

Rank modelo 3 PNO	Rank modelo 4 PNO	$\theta$ modelo 3 PNO	$\theta$ modelo 4 PNO
2284	1407	0.578	0.953
7608	8916	-0.658	-1.016

Tabela 27: Maiores diferenças de colocação

Ao analisar os 100 maiores valores de  $\theta$  no modelo de 3 parâmetros apenas 14 destes não pertence aos 100 maiores valores de  $\theta$  no modelo de 4 parâmetros, Tabela 28 apresenta

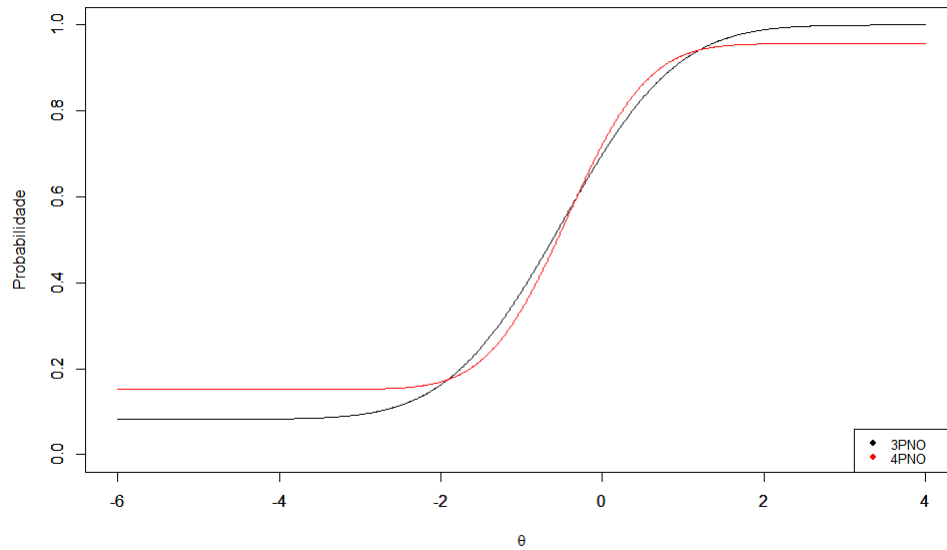


Figura 35: CCI para item 19. Estimativas três parâmetros:  $a=0.9$ ,  $b=-0.45$ ,  $c=0.08$ . Estimativas 4 parâmetros:  $a=1.28$ ,  $b=-0.55$ ,  $c=0.15$ ,  $d=0.04$

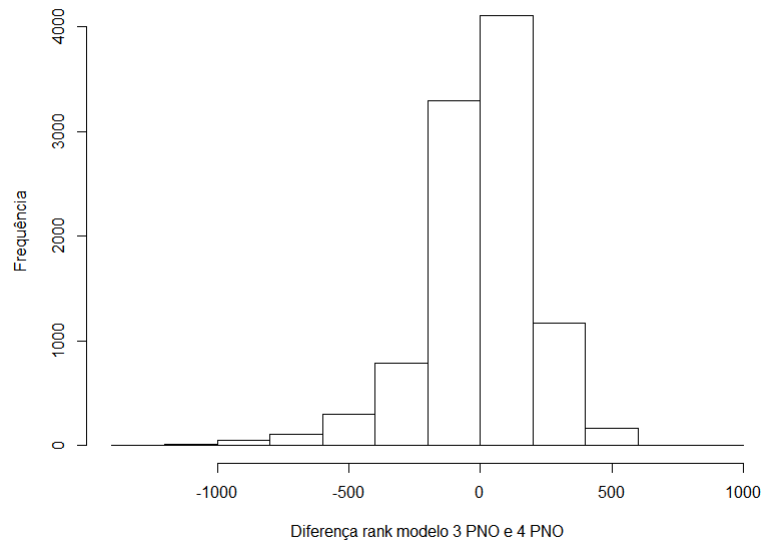


Figura 36: Diferença da colocação dos candidatos em relação ao modelo de 3 parâmetros com o modelo de 4 parâmetros

a diferença de colocação dos 14 indivíduos não pertencentes ao top 100 do modelo de 4 parâmetros. A maior diferença entre os que permaneceram entre os 100 maiores esta no individuo que foi classificado como 100º para o modelo de 3 parâmetros e como 13º para o

modelo de 4 parâmetros ( $\theta$  no modelo de 3 parâmetros de 2.21 e  $\theta$  no modelo de 4 parâmetros de 3.03). Pode se perceber então uma diferença pratica significativa ao utilizar o modelo de 4 parâmetros em relação ao de 3 parâmetros.

$\theta$ Modelo 3 PNO	$\theta$ Modelo 4 PNO	Rank modelo 3 PNO	Rank modelo 4 PNO	Diferença entre rank
2.355	2.301	59	108	49
2.337	2.282	63	116	53
2.326	2.288	67	112	45
2.316	2.328	68	105	37
2.302	2.329	72	104	32
2.296	2.249	77	124	47
2.279	2.261	82	120	38
2.269	2.283	86	114	28
2.260	2.201	91	140	49
2.243	2.155	93	154	61
2.225	2.235	95	127	32
2.220	2.297	97	111	14
2.215	2.161	98	152	54
2.211	2.331	99	103	4

Tabela 28: Diferença entre indivíduos top 100

## 6 Conclusão

Essa dissertação explorou os problemas de se fazer inferência em modelos de Teoria de Resposta ao Item, em especial o modelo de 3 parâmetros para respostas dicotômicas, sob abordagem Bayesiana via MCMC, através de comparação de algoritmos existentes para isto. Foram utilizados quatro algoritmos neste trabalho, todos algoritmos seguem uma estrutura de algoritmo Gibbs Sampling. Dois destes, Gonçalves et al. (2018) e Béguin and Glas (2001), utilizam de variáveis auxiliares para a sua construção; um considera apenas uma das variáveis auxiliares proposto por Gonçalves et al. (2018) e utiliza de passos Metropolis Hastings para estimar os parâmetros  $a$ ,  $b$ ,  $\theta$  (algoritmo Metropolis I); por último um algoritmo que não considera nenhuma variável auxiliar e utiliza apenas de metodologia Metropolis para estimar os parâmetros (algoritmo Metropolis II).

A comparação destes algoritmos se dá de forma empírica através de estudos simulados, onde foram simulado diferente bancos de dados com diferentes números de itens e indivíduos. Para amostra com menor número de itens e de indivíduos a performance dos algoritmos teve a seguinte ordem: Gonçalves et al. (2018), Metropolis II, Metropolis I e Béguin and Glas (2001). Ao aumentar o número de itens e de indivíduos, a performance do algoritmo de Béguin and Glas (2001) melhora em comparação aos demais, chegando a ser um pouco superior ao algoritmo de Gonçalves et al. (2018) para o estudo de maior amostra (10000 indivíduos e 60 itens). O algoritmo de Gonçalves et al. (2018) apresentou uma boa performance em todos os estudos.

Foi proposta também uma extensão ao algoritmo de Gonçalves et al. (2018) para o modelo de 4 parâmetros, que considera uma assintota superior para os itens de maneira que a suas probabilidades sejam inferiores a um. Foram realizadas simulações de bancos de dados com diferentes números de itens e de indivíduos, através do modelo de 4 parâmetros, e realizadas estimativas para os parâmetros utilizados para as simulações. Pode-se perceber que o algoritmo realiza estimativas dos parâmetros de maneira satisfatória.

Foi realizado um estudo comparativo da performance do algoritmo do modelo de 4 parâmetros em dados do modelo de 3 parâmetros e do algoritmo do modelo de 3 parâmetros em dados do modelo de 4 parâmetros. Percebeu-se que o algoritmo de 4 parâmetros é capaz

de realizar estimativas de maneira eficaz para o modelo de 3 parâmetros. Já o algoritmo de 3 parâmetros mostrou-se frágil para realizar estimativas para dados provenientes do modelo de 4 parâmetros.

Aplicando os algoritmos de 3 parâmetros e 4 parâmetros no banco de dados da prova de Matemática do Enem de 2017, pode-se perceber que há impacto do parâmetro de assintota superior ao banco e assim há uma pequena diferença nas estimativas para os traços latentes dos indivíduos, levando a uma leve concentração em valores abaixo de 0 para o modelo de 4 parâmetros em relação ao modelo de 3 parâmetros.

Os resultados apresentados nessa dissertação são pioneiros na comparação de algoritmos MCMC para Inferência Bayesiana em modelos de TRI e auxiliam os usuários dos modelos estudados na escolha do algoritmo a ser utilizado. Como trabalhos futuros, podem ser consideradas comparações entre os algoritmos propostos para o modelo de 4 parâmetros, para detectar se o comportamento observado neste estudo permaneceria, e comparação de algoritmos de modelos mais gerais, como modelos politômicos e modelos multidimensionais.

## Referências

- J. H. Albert. Bayesian estimation of normal ogive item response curves using gibbs sampling. *Journal of educational statistics*, 17(3):251–269, 1992.
- M. A. Barton and F. M. Lord. An upper asymptote for the three-parameter logistic item-response model. *ETS Research Report Series*, 1981(1):i–8, 1981.
- A. A. Béguin and C. A. Glas. Mcmc estimation and some model-fit analysis of multidimensional irt models. *Psychometrika*, 66(4):541–561, 2001.
- A. Binet and T. Simon. Méthodes nouvelles pour le diagnostic du niveau intellectuel des anormaux. *L'année Psychologique*, 11(1):191–244, 1904.
- A. Birnbaum. Some latent trait models and their use in inferring an examinee's ability. *Statistical theories of mental test scores*, 1968.
- S. A. Culpepper. Revisiting the 4-parameter item response model: Bayesian estimation and application. *psychometrika*, 81(4):1142–1163, 2016.
- A. Gelman, D. B. Rubin, et al. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.
- S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.
- F. B. Gonçalves, B. Dias, and T. Machado Soares da Costa Campos. Bayesian item response model: a generalized approach for the abilities' distribution using mixtures. *Journal of Statistical Computation and Simulation*, 88(5):967–981, 2018.
- W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.
- F. Lord. A theory of test scores. *Psychometric monographs*, 1952.



- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6): 1087–1092, 1953.
- K. W. Ng, G.-L. Tian, and M.-L. Tang. *Dirichlet and related distributions: Theory, methods and applications*, volume 888. John Wiley & Sons, 2011.
- G. Rash. Probabilistic models for some intelligence and attainment tests. *Copenhagen: Danish Institute for Educational Research*, 1960.
- G. O. Roberts, A. Gelman, W. R. Gilks, et al. Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997.
- C. Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101, 1904.
- L. L. Thurstone. A method of scaling psychological and educational tests. *Journal of educational psychology*, 16(7):433, 1925.

# A Cálculos

## A.1 Cálculos Algoritmo Béguin and Glas (2001)

Cálculo da condicional completa para  $\Psi_1$ , devem ser considerado dois casos, sendo estes quando  $Y_{ij} = 1$  e quando  $Y_{ij} = 0$ . Para  $Y_{ij} = 1$  tem-se:

$$\begin{aligned}
 \pi(Z_{ij}|\cdot) &\propto \pi(Y_{ij} = 1|Z_{ij})\pi(Z_{ij}) \\
 &\propto (1I_{\{Z_{ij}=1\}} + c_i I_{\{Z_{ij}=0\}})(\Phi(a_i\theta_j - b_i))^{Z_{ij}}(1 - \Phi(a_i\theta_j - b_i))^{1-Z_{ij}} \\
 &\propto \Phi(a_i\theta_j - b_i)I_{\{Z_{ij}=1\}} + c_i(1 - \Phi(a_i\theta_j - b_i))I_{\{Z_{ij}=0\}} \\
 &= \frac{\Phi(a_i\theta_j - b_i)}{\Phi(a_i\theta_j - b_i) + c_i(1 - \Phi(a_i\theta_j - b_i))}I_{\{Z_{ij}=1\}} + \frac{c_i(1 - \Phi(a_i\theta_j - b_i))}{\Phi(a_i\theta_j - b_i) + c_i(1 - \Phi(a_i\theta_j - b_i))}I_{\{Z_{ij}=0\}}.
 \end{aligned}$$

Assim é possível perceber que  $(Z_{ij}|Y_{ij} = 1, \cdot) \sim \text{Bernoulli}\left(\frac{\Phi(a_i\theta_j - b_i)}{\Phi(a_i\theta_j - b_i) + c_i(1 - \Phi(a_i\theta_j - b_i))}\right)$ .

Para  $Y_{ij} = 0$  tem-se:

$$\begin{aligned}
 \pi(Z_{ij}|\cdot) &\propto \pi(Y_{ij} = 0|Z_{ij})\pi(Z_{ij}) \\
 &\propto (0I_{\{Z_{ij}=1\}} + (1 - c_i)I_{\{Z_{ij}=0\}})(\Phi(a_i\theta_j - b_i))^{Z_{ij}}(1 - \Phi(a_i\theta_j - b_i))^{1-Z_{ij}} \\
 &\propto (1 - c_i)I_{\{Z_{ij}=0\}}(1 - \Phi(a_i\theta_j - b_i)).
 \end{aligned}$$

Logo, percebe-se que  $P(Z_{ij} = 0|Y_{ij} = 0, \cdot) = 1$ , assim  $(Z_{ij}|Y_{ij} = 0, \cdot)$  é um ponto de massa no zero.

Para  $\Psi_2$  tem-se:

$$\begin{aligned}
\pi(\theta_j|\cdot) &\propto \prod_{i=1}^I \pi(X_{ij}|a_i, b_i, \theta_j) \pi(\theta_j) \\
&\propto \prod_{i=1}^I e^{-\frac{(X_{ij}-a_i\theta_j+b_i)^2}{2}} e^{-\frac{\theta_j^2}{2}} \\
&\propto \exp \left\{ -\frac{\left(-2\theta_j \sum_{i=1}^I X_{ij}a_i + \sum_{i=1}^I a_i^2\theta_j^2 - 2\theta_j \sum_{i=1}^I a_ib_i\right)}{2} - \frac{\theta_j^2}{2} \right\} \\
&\propto \exp \left\{ -\frac{\theta_j^2 \left(\sum_{i=1}^I a_i^2 + 1\right) - 2\theta_j \left(\sum_{i=1}^I a_i(X_{ij} + b_i)\right)}{2} \right\} \\
&\propto \exp \left\{ -\frac{\left(\sum_{i=1}^I a_i^2 + 1\right) \left(\theta_j^2 - 2\theta_j \frac{\sum_{i=1}^I a_i(X_{ij}+b_i)}{\sum_{i=1}^I a_i^2+1}\right)}{2} \right\},
\end{aligned}$$

assim,  $(\theta_j|\cdot) \sim N\left(\frac{\sum_{i=1}^I a_i(X_{ij}+b_i)}{\sum_{i=1}^I a_i^2+1}, \frac{1}{\sum_{i=1}^I a_i^2+1}\right)$ .

Para  $\Psi_3$  tem-se:

$$\begin{aligned}
\pi(a_i, b_i|\cdot) &\propto \prod_{j=1}^J \pi(X_{ij}|a_i, b_i, \theta_j) \pi(a_i, b_i) I_{a_i>0} \\
&\propto \prod_{j=1}^J \exp \left\{ -\frac{1}{2} \left( (X_{ij} - a_i\theta_j + b_i)^2 + \frac{(a_i - \mu_a)^2}{\sigma_a^2} + \frac{(b_i - \mu_b)^2}{\sigma_b^2} \right) \right\} I_{a_i>0} \\
&\propto \exp \left\{ -\frac{1}{2} \left( \sum_{j=1}^J (X_{ij} - a_i\theta_j + b_i)^2 + \frac{(a_i - \mu_a)^2}{\sigma_a^2} + \frac{(b_i - \mu_b)^2}{\sigma_b^2} \right) \right\} I_{a_i>0} \\
&\propto \exp \left\{ -\frac{1}{2} \left( -2a_i \sum_{j=1}^J X_{ij}\theta_j + a_i^2 \sum_{j=1}^J \theta_j^2 - 2a_ib_i \sum_{j=1}^J \theta_j + Jb_i^2 + 2b_i \sum_{j=1}^J X_{ij} \right. \right. \\
&\quad \left. \left. + \frac{a_i^2}{\sigma_a^2} - \frac{2a_i\mu_a}{\sigma_a^2} + \frac{b_i^2}{\sigma_b^2} - \frac{2\mu_b b_i}{\sigma_b^2} \right) \right\} I_{a_i>0} \\
&\propto \exp \left\{ -\frac{1}{2} \left( a_i^2 \left( \sum_{j=1}^J \theta_j^2 + \frac{1}{\sigma_a^2} \right) + b_i^2 \left( J + \frac{1}{\sigma_b^2} \right) + 2a_ib_i \sum_{j=1}^J \theta_j \right. \right. \\
&\quad \left. \left. - 2a_i \left( \sum_{j=1}^J X_{ij}\theta_j + \frac{\mu_a}{\sigma_a^2} \right) - 2b_i \left( -\sum_{j=1}^J X_{ij} + \frac{\mu_b}{\sigma_b^2} \right) \right) \right\} I_{a_i>0}.
\end{aligned}$$

Podemos reescrever da seguinte maneira:

$$\pi(a_i, b_i | \cdot) \propto \exp \left\{ -\frac{1}{2} \left( \frac{a_i^2}{(1-\gamma^2)(\sigma_a^{2*})} + \frac{b_i^2}{(1-\gamma^2)(\sigma_b^{2*})} - \frac{2\gamma a_i b_i}{(1-\gamma^2)\sigma_a^* \sigma_b^*} \right. \right. \\ \left. \left. - 2 \frac{a_i}{(1-\gamma^2)} \left( \frac{\mu_a^*}{\sigma_a^{2*}} - \frac{\gamma \mu_b^*}{\sigma_a^* \sigma_b^*} \right) - \frac{2b_i}{(1-\gamma^2)} \left( \frac{\mu_b^*}{\sigma_b^{2*}} - \frac{\gamma \mu_a^*}{\sigma_a^* \sigma_b^*} \right) \right\} I_{a_i > 0}.$$

Onde:

$$\sigma_a^{2*} = \frac{\sigma_a^2}{(\sigma_a^2 \sum_{j=1}^J \theta_j^2 + 1)(1-\gamma^2)},$$

$$\sigma_b^{2*} = \frac{\sigma_b^2}{(\sigma_b^2 J + 1)(1-\gamma^2)},$$

$$\gamma = \frac{\sigma_a \sigma_b \sum_{j=1}^J \theta_j}{[(\sigma_a^2 \sum_{j=1}^J \theta_j^2 + 1)(\sigma_b^2 J + 1)]^{1/2}},$$

$$\mu_a^* = \sigma_a^{2*} \left( \sum_{j=1}^J (X_{ij} \theta_j) + \mu_a \sigma_a^{-2} \right) - \sigma_a^* \sigma_b^* \gamma \left( \sum_{j=1}^J X_{ij} - \mu_b \sigma_b^{-2} \right),$$

$$\mu_b^* = \sigma_a^* \sigma_b^* \gamma \left( \sum_{j=1}^J (X_{ij} \theta_j) + \mu_a \sigma_a^{-2} \right) - \sigma_b^{2*} \left( \sum_{j=1}^J X_{ij} - \mu_b \sigma_b^{-2} \right).$$

Assim:

$$(a_i, b_i | \cdot) \sim N_2(\mu, \Sigma).$$

$$\mu = \begin{bmatrix} \mu_a^* \\ \mu_b^* \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma_a^{2*} & \sigma_a \sigma_b \gamma \\ \sigma_a^* \sigma_b^* \gamma & \sigma_b^{2*} \end{bmatrix}$$

Para  $\Psi_4$  tem-se:

$$\pi(c_i | \cdot) \propto \prod_{j=1}^J \pi(Y_{ij} | Z_{ij}, c_i) \pi(c_i) \\ \propto \prod_{j=1}^J (c_i^{Y_{ij}} (1-c_i)^{1-Y_{ij}} I_{\{Z_{ij}=0\}} + I_{\{Z_{ij}=1\}}) c_i^{\alpha c - 1} (1-c_i)^{\beta c - 1} \\ \propto c_i^{\sum_{j|Z_{ij}=0} Y_{ij}} (1-c_i)^{\sum_{j=1}^J I_{\{Z_{ij}=0\}} - \sum_{j|Z_{ij}=1} Y_{ij}} c_i^{\alpha c - 1} (1-c_i)^{\beta c - 1} \\ \propto c_i^{\sum_{j|Z_{ij}=0} Y_{ij} + \alpha c - 1} (1-c_i)^{\sum_{j=1}^J I_{\{Z_{ij}=0\}} - \sum_{j|Z_{ij}=1} Y_{ij} + \beta c - 1},$$

$$\text{assim } (c_i | \cdot) \sim \text{Beta}(\sum_{j|Z_{ij}=0} Y_{ij} + \alpha, \sum_{j=1}^J I_{\{Z_{ij}=0\}} - \sum_{j|Z_{ij}=1} Y_{ij} + \beta c)$$

## A.2 Cálculos Algoritmo Gonçalves et al. (2018)

Para condicional completa de  $\Psi_1$ , deve-se ser considerados dois casos. Primeiro para  $Y_{ij} = 1$ , tem-se que:

$$\begin{aligned}
\pi(Z_{ij}, X_{ij} | Y_{ij} = 1) &\propto \pi(Y_{ij} = 1 | Z_{ij}, X_{ij}) \pi(Z_{ij} | c_i) \pi(X_{ij} | a_i, b_i, \theta_j) \\
&\propto (I_{Z_{ij}=1} I_{X_{ij}=0} + I_{Z_{ij}=0} I_{X_{ij}>0}) \pi(Z_{ij} | c_i) \pi(X_{ij} | a_i, b_i, \theta_j) \\
&\propto \pi(Z_{ij} | c_i) p_i(X_{ij} | a_i, b_i, \theta_j) I_{Z_{ij}=1} I_{X_{ij}=0} + \pi(Z_{ij} | c_i) \pi(X_{ij} | a_i, b_i, \theta_j) I_{Z_{ij}=0} I_{X_{ij}>0} \\
&\propto c_i I_{Z_{ij}=1} I_{X_{ij}=0} + (1 - c_i) \phi(x_{ij} - (a_i \theta_j - b_i)) I_{Z_{ij}=0} I_{X_{ij}>0} \\
&\propto c_i I_{Z_{ij}=1} I_{X_{ij}=0} + \Phi(a_i \theta_j - b_i) (1 - c_i) \frac{\phi(x_{ij} - (a_i \theta_j - b_i))}{\Phi(a_i \theta_j - b_i)} I_{Z_{ij}=0} I_{X_{ij}>0}.
\end{aligned}$$

Assim  $(Z_{ij} | Y_{ij} = 1, \cdot) \sim \text{Bernoulli} \left( \frac{c_i}{c_i + (1 - c_i) \Phi(a_i \theta_j - b_i)} \right)$  e  $(X_{ij} | Z_{ij} = 1, Y_{ij} = 1, \cdot)$  é um ponto de massa em zero,  $(X_{ij} | Z_{ij} = 1, Y_{ij} = 1, \cdot) \sim N_{(0, \infty)}(a_i \theta_j - b_i, 1)$ .

Para  $Y_{ij} = 0$  tem-se que:

$$\begin{aligned}
\pi(Z_{ij}, X_{ij} | Y_{ij} = 0, \cdot) &\propto \pi(Y_{ij} = 0 | Z_{ij}, X_{ij}) \pi(Z_{ij} | c_i) \pi(X_{ij} | a_i, b_i, \theta_j) \\
&\propto (I_{Z_{ij}=0} I_{X_{ij}<0}) \pi(Z_{ij} | c_i) \pi(X_{ij} | a_i, b_i, \theta_j).
\end{aligned}$$

Assim  $(Z_{ij} | Y_{ij} = 0, \cdot)$  é um ponto de massa em zero e  $(X_{ij} | Y_{ij} = 0, Z_{ij} = 0, \cdot) \sim N_{(-\infty, 0)}(a_i \theta_j - b_i, 1)$ .

Para  $\Psi_2$ , como  $(X_{ij} | Z_{ij} = 0, a_i, b_i, \theta_j) \sim N(a_i \theta_j - b_i, 1)$ , tem-se que:

$$\begin{aligned}
\pi(\theta_j | \cdot) &\propto \prod_{i; Z_{ij}=0} \pi(X_{ij} | a_i, b_i, \theta_j) \pi(\theta_j) \\
&\propto \prod_{i; Z_{ij}=0} e^{-\frac{(X_{ij} - a_i \theta_j + b_i)^2}{2}} e^{-\frac{\theta_j^2}{2}} \\
&\propto \exp \left\{ -\frac{\left( -2\theta_j \sum_{i; Z_{ij}=0} X_{ij} a_i + \sum_{i; Z_{ij}=0} a_i^2 \theta_j^2 - 2\theta_j \sum_{i; Z_{ij}=0} a_i b_i \right) - \frac{\theta_j^2}{2}}{2} \right\} \\
&\propto \exp \left\{ -\frac{\theta_j^2 \left( \sum_{i; Z_{ij}=0} a_i^2 + 1 \right) - 2\theta_j \left( \sum_{i; Z_{ij}=0} a_i (X_{ij} + b_i) \right)}{2} \right\} \\
&\propto \exp \left\{ -\frac{\left( \sum_{i; Z_{ij}=0} a_i^2 + 1 \right) \left( \theta_j^2 - 2\theta_j \frac{\sum_{i; Z_{ij}=0} a_i (X_{ij} + b_i)}{\sum_{i=1}^I a_i^2 + 1} \right)}{2} \right\},
\end{aligned}$$

$$\text{assim } (\theta_j | \cdot) \sim N \left( \frac{\sum_{i; Z_{ij}=0} a_i (X_{ij} + b_i)}{\sum_{i; Z_{ij}=0} a_i^2 + 1}, \frac{1}{\sum_{i; Z_{ij}=0} a_i^2 + 1} \right).$$

Para  $\Psi_3$  tem-se:

$$\begin{aligned} \pi(a_i, b_i | \cdot) &\propto \prod_{j; Z_{ij}=0} \pi(X_{ij} | a_i, b_i, \theta_j) \pi(a_i, b_i) I_{a_i > 0} \\ &\propto \prod_{j; Z_{ij}=0} \exp \left\{ -\frac{1}{2} \left( (X_{ij} - a_i \theta_j + b_i)^2 + \frac{(a_i - \mu_a)^2}{\sigma_a^2} + \frac{(b_i - \mu_b)^2}{\sigma_b^2} \right) \right\} I_{a_i > 0} \\ &\propto \exp \left\{ -\frac{1}{2} \left( \sum_{j; Z_{ij}=0} (X_{ij} - a_i \theta_j + b_i)^2 + \frac{(a_i - \mu_a)^2}{\sigma_a^2} + \frac{(b_i - \mu_b)^2}{\sigma_b^2} \right) \right\} I_{a_i > 0} \\ &\propto \exp \left\{ -\frac{1}{2} \left( -2a_i \sum_{j; Z_{ij}=0} X_{ij} \theta_j + a_i^2 \sum_{j; Z_{ij}=0} \theta_j^2 - 2a_i b_i \sum_{j=1}^J \theta_j + N_J b_i^2 + 2b_i \sum_{j=1}^J X_{ij} \right. \right. \\ &\quad \left. \left. + \frac{a_i^2}{\sigma_a^2} - \frac{2a_i \mu_a}{\sigma_a^2} + \frac{b_i^2}{\sigma_b^2} - \frac{2\mu_b b_i}{\sigma_b^2} \right) \right\} I_{a_i > 0} \\ &\propto \exp \left\{ -\frac{1}{2} \left( a_i^2 \left( \sum_{j; Z_{ij}=0} \theta_j^2 + \frac{1}{\sigma_a^2} \right) + b_i^2 \left( N_J + \frac{1}{\sigma_b^2} \right) + 2a_i b_i \sum_{j; Z_{ij}=0} \theta_j \right. \right. \\ &\quad \left. \left. - 2a_i \left( \sum_{j; Z_{ij}=0} X_{ij} \theta_j + \frac{\mu_a}{\sigma_a^2} \right) - 2b_i \left( - \sum_{j; Z_{ij}=0} X_{ij} + \frac{\mu_b}{\sigma_b^2} \right) \right) \right\} I_{a_i > 0}. \end{aligned}$$

$$\text{Onde } N_J = \sum_{j=1}^J I_{Z_{ij}=0}.$$

Podemos reescrever da seguinte maneira:

$$\begin{aligned} \pi(a_i, b_i | \cdot) &\propto \exp \left\{ -\frac{1}{2} \left( \frac{a_i^2}{(1 - \gamma^2)(\sigma_a^{2*})} + \frac{b_i^2}{(1 - \gamma^2)(\sigma_b^{2*})} - \frac{2\gamma a_i b_i}{(1 - \gamma^2)\sigma_a^* \sigma_b^*} \right. \right. \\ &\quad \left. \left. - 2 \frac{a_i}{(1 - \gamma^2)} \left( \frac{\mu_a^*}{\sigma_a^{2*}} - \frac{\gamma \mu_b^*}{\sigma_a^* \sigma_b^*} \right) - \frac{2b_i}{(1 - \gamma^2)} \left( \frac{\mu_b^*}{\sigma_b^{2*}} - \frac{\gamma \mu_a^*}{\sigma_a^* \sigma_b^*} \right) \right) \right\}. \end{aligned}$$

Onde:

$$\begin{aligned} \sigma_a^{2*} &= \frac{\sigma_a^2}{(\sigma_a^2 \sum_{j; Z_{ij}=0} \theta_j^2 + 1)(1 - \gamma^2)}, \\ \sigma_b^{2*} &= \frac{\sigma_b^2}{(\sigma_b^2 N_J + 1)(1 - \gamma^2)}, \\ \gamma &= \frac{\sigma_a \sigma_b \sum_{j; Z_{ij}=0} \theta_j}{[(\sigma_a^2 \sum_{j; Z_{ij}=0} \theta_j^2 + 1)(\sigma_b^2 N_J + 1)]^{1/2}}, \end{aligned}$$

$$\mu_a^* = \sigma_a^{2*} \left( \sum_{j; Z_{ij}=0} (X_{ij}\theta_j) + \mu_a \sigma_a^{-2} \right) - \sigma_a^* \sigma_b^* \gamma \left( \sum_{j; Z_{ij}=0} X_{ij} - \mu_b \sigma_b^{-2} \right),$$

$$\mu_b^* = \sigma_a^* \sigma_b^* \gamma \left( \sum_{j; Z_{ij}=0} (X_{ij}\theta_j) + \mu_a \sigma_a^{-2} \right) - \sigma_b^{2*} \left( \sum_{j; Z_{ij}=0} X_{ij} - \mu_b \sigma_b^{-2} \right).$$

Assim:

$$(a_i, b_i | \cdot) \sim N_2(\mu, \Sigma) I_{a_i > 0}.$$

$$\mu = \begin{bmatrix} \mu_a^* \\ \mu_b^* \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma_a^{2*} & \sigma_a^* \sigma_b^* \gamma \\ \sigma_a^* \sigma_b^* \gamma & \sigma_b^{2*} \end{bmatrix}$$

Para  $\Psi_4$  tem-se que:

$$\begin{aligned} \pi(c_i | \cdot) &\propto \prod_{j=1}^J \pi(Z_{ij} | c_i) \pi(c_i) \\ &\propto \prod_{j=1}^J (c_i)^{Z_{ij}} (1 - c_i)^{1 - Z_{ij}} c_i^{\alpha_c - 1} (1 - c_i)^{\beta_c - 1} \\ &\propto c_i^{\sum_{j=1}^J Z_{ij}} (1 - c_i)^{J - \sum_{j=1}^J Z_{ij}} c_i^{\alpha_c - 1} (1 - c_i)^{\beta_c - 1} \\ &\propto c_i^{\sum_{j=1}^J Z_{ij} + \alpha_c - 1} (1 - c_i)^{J - \sum_{j=1}^J Z_{ij} + \beta_c - 1}, \end{aligned}$$

$$\text{assim } (c_i | \cdot) \sim \text{Beta}(\sum_{j=1}^J Z_{ij} + \alpha_c, J - \sum_{j=1}^J Z_{ij} + \beta_c).$$

### A.3 Cálculos Algoritmo Metropolis-Hastings I

Para a condicional completa de  $Z_{ij}$  temos que considerar dois casos. Primeiro, quando  $Y_{ij} = 1$ :

$$\begin{aligned} \pi(Z_{ij} | \cdot) &\propto \pi(Y_{ij} = 1 | Z_{ij}, a_i, b_i, \theta_j) \pi(Z_{ij} | c_i) \\ &\propto (I_{Z_{ij}=1} + \Phi(a_i \theta_j - b_i) I_{Z_{ij}=0}) (c_i^{Z_{ij}} (1 - c_i)^{1 - Z_{ij}}) \\ &\propto c_i I_{Z_{ij}=1} + (1 - c_i) \Phi(a_i \theta_j - b_i) I_{Z_{ij}=0}. \end{aligned}$$

Assim, é possível perceber que  $(Z_{ij} | Y_{ij} = 1) \sim \text{Bernoulli} \left( \frac{c_i}{c_i + (1 - c_i) \Phi(a_i \theta_j - b_i)} \right)$ . E por construção é possível perceber que  $Z_{ij}$  é um ponto de massa no zero quando  $Y_{ij} = 0$ .

Para gerar a probabilidade de aceitação de  $\psi_2$ , tem-se que a distribuição alvo é dada por:

$$\begin{aligned}\pi(\theta_j|\cdot) &\propto \prod_{i;Z_{ij}=0} \pi(Y_{ij}|a_i, b_i, \theta_j)\pi(\theta_j) \\ &\propto \prod_{i;Z_{ij}=0} p_i(\theta_j)^{Y_{ij}}(1-p_i(\theta_j))^{1-Y_{ij}}e^{-\frac{\theta_j^2}{2}}.\end{aligned}$$

Onde  $p_i(\theta) = \Phi(a_i\theta - b_i)$ . A distribuição proposta para a geração desta distribuição é a distribuição Normal, logo uma função par, assim, utilizando equação (3), para se calcular a probabilidade de aceitação, temos que:

$$\begin{aligned}\alpha(\theta_j^*, \theta_j^{k-1}) &= \min\left(1, \frac{\pi(\theta_j^*)}{\pi(\theta_j^{k-1})}\right) \\ &= \min\left(1, \exp\{-0.5(\theta_j^{*2} - \theta_j^{k-12})\} \prod_{i;Z_{ij}=0} \left[\frac{p_i(\theta_j^*)}{p_i(\theta_j^{k-1})}\right]^{Y_{ij}} \left[\frac{1-p_i(\theta_j^*)}{1-p_i(\theta_j^{k-1})}\right]^{1-Y_{ij}}\right).\end{aligned}$$

Para gerar a probabilidade de aceitação de  $\psi_3$ , tem-se que a distribuição alvo é dado por:

$$\begin{aligned}\pi(a_i, b_i|\cdot) &\propto \prod_{j;Z_{ij}=0} \pi(Y_{ij}|a_i, b_i, \theta_j)\pi(a_i, b_i)I_{a_i>0} \\ &\propto \prod_{i;Z_{ij}=0} p_j(a_i, b_i)^{Y_{ij}}(1-p_j(\theta_j))^{1-Y_{ij}}e^{-\frac{(a_i-\mu_a)^2}{2\sigma_a^2} - \frac{(b_i-\mu_b)^2}{2\sigma_b^2}} I_{a_i>0},\end{aligned}$$

onde  $p_j(a_i, b_i) = \phi(a_i\theta_j - b_i)$ , a distribuição proposta é uma distribuição Normal bivariada com correlação zero. Assim a probabilidade de aceitação é:

$$\begin{aligned}\alpha((a_i^*, b_i^*), (a_i^{k-1}, b_i^{k-1})) &= \min\left(1, \frac{\pi(a_i^*, b_i^*|\cdot)}{\pi(a_i^{k-1}, b_i^{k-1}|\cdot)} I_{a_i>0}\right) \\ &= \min\left(1, \exp\left\{-\frac{1}{2\sigma_a^2}(a_i^{*2} - a_i^{k-12} - 2\mu_a(a_i^* - a_i^{k-1}))\right\}\right. \\ &\quad \left.\exp\left\{-\frac{1}{2\sigma_b^2}(b_i^{*2} - b_i^{k-12} - 2\mu_b(b_i^* - b_i^{k-1}))\right\}\right. \\ &\quad \left.\prod_{j;Z_{ij}=0} \left[\frac{p_j(a_i^*, b_i^*)}{p_j(a_i^{k-1}, b_i^{k-1})} \frac{1-p_j(a_i^*, b_i^*)}{1-p_j(a_i^{k-1}, b_i^{k-1})}\right] I_{a_i>0}\right).\end{aligned}$$

Para  $\psi_4$  tem-se que a sua distribuição condicional completa é idêntica a construída para o algoritmo de Gonçalves et al. (2018), ver A.2.



## A.4 Cálculos Algoritmo Metropolis-Hastings II

Para os cálculos de densidade de condicional completa de  $\psi_1$  e  $\psi_2$  checar A.3. Para o cálculo de  $\psi_3$  tem-se que a sua distribuição alvo é:

$$\begin{aligned}\pi(c_i|\cdot) &\propto \prod_{j=1}^J \pi(Y_{ij}|a_i, b_i, c_i, \theta_j) \pi(c_i) \\ &\propto \prod_{j=1}^J p_{ij}(c_i)^{Y_{ij}} (1 - p_{ij}(c_i))^{1-Y_{ij}} c_i^{\alpha c-1} (1 - c_i)^{\beta c-1}.\end{aligned}$$

A distribuição proposta utilizada é  $h(c_i^*, c_i^{k-1}) \sim \text{Uniforme}(\psi(c_i^{k-1}), \gamma(c_i^{k-1}))$ , onde  $\psi(c_i^{k-1}) = \max(0, c_i^{k-1} - \delta)$  e  $\gamma(c_i^{k-1}) = \min(1, c_i^{k-1} + \delta)$ . Logo a probabilidade de aceitação é dada por:

$$\begin{aligned}\alpha(c_i^{k-1}, c_i^*) &= \min \left( 1, \frac{\pi(c_i^*|\cdot) h(c_i^*, c_i^{k-1})}{\pi(c_i^{k-1}|\cdot) h(c_i^{k-1}, c_i^*)} \right) \\ &= \min \left( 1, \prod_{j=1}^J \left[ \frac{p_{ij}(c_i^*)}{p_{ij}(c_i^{k-1})} \right]^{Y_{ij}} \left[ \frac{1 - p_{ij}(c_i^*)}{1 - p_{ij}(c_i^{k-1})} \right]^{1-Y_{ij}} \frac{(c_i^*)^{\alpha c-1} (1 - c_i^*)^{\beta c-1}}{(c_i^{k-1})^{\alpha c-1} (1 - c_i^{k-1})^{\beta c-1}} \frac{\gamma(c_i^*) - \psi(c_i^*)}{\gamma(c_i^{k-1}) - \psi(c_i^{k-1})} \right)\end{aligned}$$

## B Cálculos Algoritmo Modelo de 4 parâmetros

Para condicional completa de  $\Psi_1$ , deve-se ser considerados dois casos. Primeiro para  $Y_{ij} = 1$ , tem-se que:

$$\begin{aligned}\pi(Z_{ij}, X_{ij}|Y_{ij} = 1) &\propto \pi(Y_{ij} = 1|Z_{ij}, X_{ij}) \pi(Z_{ij}|c_i, d_i) \pi(X_{ij}|a_i, b_i, \theta_j) \\ &\propto (I_{Z_{ij}=(1,0,0)} I_{X_{ij}=0} + I_{Z_{ij}=(0,1,0)} I_{X_{ij}>0}) \pi(Z_{ij}|c_i, d_i) \pi(X_{ij}|a_i, b_i, \theta_j) \\ &\propto c_i I_{Z_{ij}=(0,1,0)} I_{X_{ij}=0} + (1 - c_i - d_i) \phi(x_{ij} - (a_i \theta_j - b_i)) I_{Z_{ij}=(0,1,0)} I_{X_{ij}>0} \\ &\propto c_i I_{Z_{ij}=(1,0,0)} I_{X_{ij}=0} + \Phi(a_i \theta_j - b_i) (1 - c_i - d_i) \frac{\phi(x_{ij} - (a_i \theta_j - b_i))}{\Phi(a_i \theta_j - b_i)} I_{Z_{ij}=(0,1,0)} I_{X_{ij}>0}.\end{aligned}$$

Utilizando de uma variável auxiliar  $w_{ij} \sim \text{Bernoulli} \left( \frac{(1-d_i-c_i)\Phi(a_i\theta_j-b_i)}{c_i+(1-d_i-c_i)\Phi(a_i\theta_j-b_i)} \right)$ , tem-se que  $Z_{ij} = (1, 0, 0)$  se  $w_{ij} = 0$ ,  $Z_{ij} = (0, 1, 0)$  se  $w_{ij} = 1$  e  $X_{ij} \sim N_{(0,\infty)}(a_i\theta_j - b_i, 1)$  se  $Z_{ij} = (0, 1, 0)$ .

Para o segundo caso,  $Y_{ij} = 0$  tem-se que:

$$\begin{aligned}
\pi(Z_{ij}, X_{ij} | Y_{ij} = 0) &\propto \pi(Y_{ij} = 0 | Z_{ij}, X_{ij}) \pi(Z_{ij} | c_i, d_i) \pi(X_{ij} | a_i, b_i, \theta_j) \\
&\propto (I_{Z_{ij}=(0,0,1)} I_{X_{ij}=0} + I_{Z_{ij}=(0,1,0)} I_{X_{ij}<0}) \pi(Z_{ij} | c_i, d_i) \pi(X_{ij} | a_i, b_i, \theta_j) \\
&\propto d_i I_{Z_{ij}=(0,0,1)} I_{X_{ij}=0} + (1 - c_i - d_i) \phi(x_{ij} - (a_i \theta_j - b_i)) I_{Z_{ij}=(0,1,0)} I_{X_{ij}<0} \\
&\propto d_i I_{Z_{ij}=(0,0,1)} I_{X_{ij}=0} + (1 - \Phi(a_i \theta_j - b_i)) (1 - c_i - d_i) \frac{\phi(x_{ij} - (a_i \theta_j - b_i))}{1 - \Phi(a_i \theta_j - b_i)} I_{Z_{ij}=(0,1,0)} I_{X_{ij}<0}.
\end{aligned}$$

Utilizando uma variável auxiliar  $v_{ij} \sim \text{Bernoulli} \left( \frac{d_i}{d_i + (1 - c_i - d_i)(1 - \Phi(a_i \theta_j - b_i))} \right)$ , tem-se que  $Z_{ij} = (0, 0, 1)$  se  $v_{ij} = 1$  e  $Z_{ij} = (0, 1, 0)$  se  $v_{ij} = 0$ , e  $X_{ij} = N_{(-\infty, 0)}(a_i \theta_j - b_i, 1)$  se  $Z_{ij} = (0, 1, 0)$ .

Para  $\psi_2$  e  $\psi_3$  ver A.2. Para  $\psi_4$  temos que:

$$\begin{aligned}
\pi(c_i, 1 - c_i - d_i, d_i | \cdot) &\propto \pi(Z_{ij} | c_i, d_i) \pi(c_i, 1 - c_i - d_i, d_i) \\
&\propto \prod_{j=1}^J c_i^{Z_{ij1}} (1 - c_i - d_i)^{Z_{ij2}} (d_i)^{Z_{ij3}} c_i^{\alpha_1} (1 - c_i - d_i)^{\alpha_2} d_i^{\alpha_3} \\
&\propto c_i^{\alpha_1 + \sum_{j=1}^J Z_{ij1}} (1 - c_i - d_i)^{\alpha_2 + \sum_{j=1}^J Z_{ij2}} d_i^{\alpha_3 + \sum_{j=1}^J Z_{ij3}}.
\end{aligned}$$

Logo:

$$((c_i, 1 - c_i - d_i, d_i) | \cdot) \sim \text{Dir} \left( \alpha_1 + \sum_{j=1}^J Z_{ij1}, \alpha_2 + \sum_{j=1}^J Z_{ij2}, \alpha_3 + \sum_{j=1}^J Z_{ij3} \right).$$

# C Cadeias de Markov

## C.1 Estudo Simulado 1

Para melhor visualização dos gráficos das cadeias da função log densidade *posteriori* foram ignoradas as 10 primeiras iterações. Valor inicial das cadeias: -5747.636, -4591.284, -5909.597 e -4619.654.

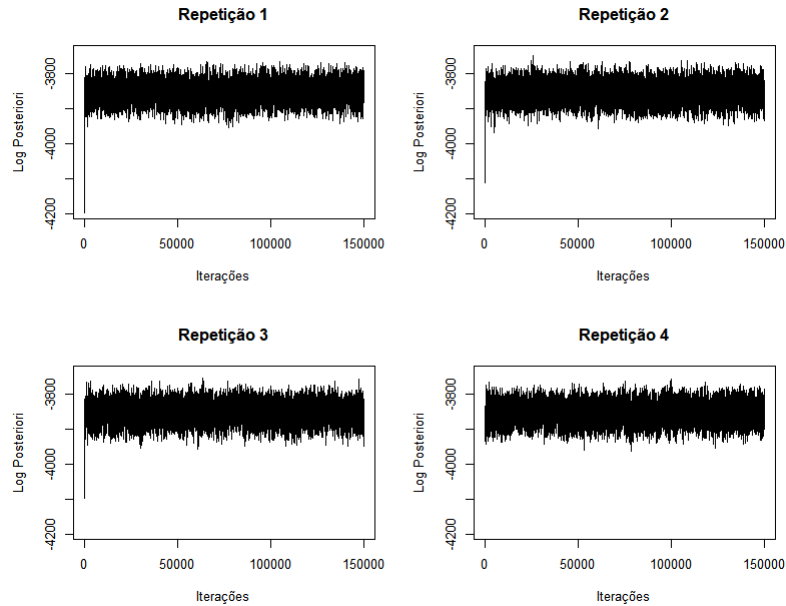


Figura 37: Cadeias Log *Posteriori* algoritmo Béguin and Glas (2001) para Estudo Simulado 1.

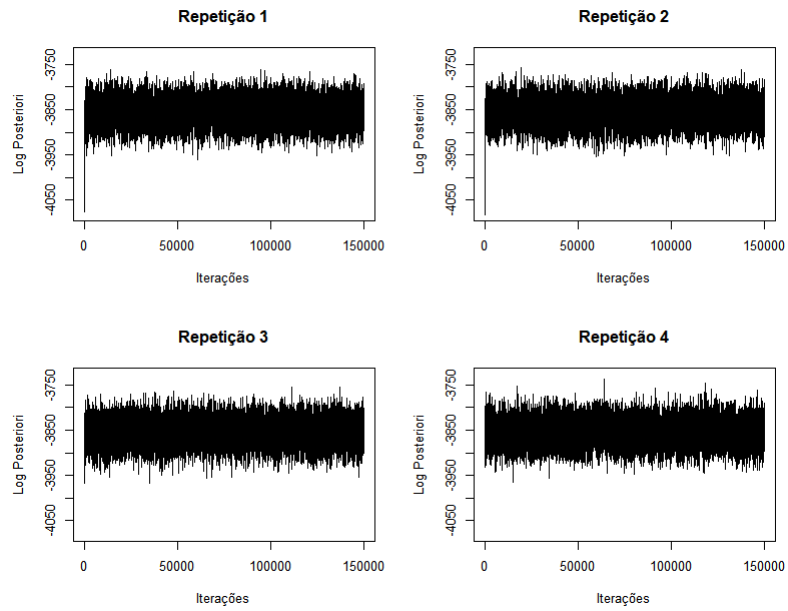


Figura 38: Cadeias Log *Posteriori* algoritmo Gonçalves et al. (2018) para Estudo Simulado 1.

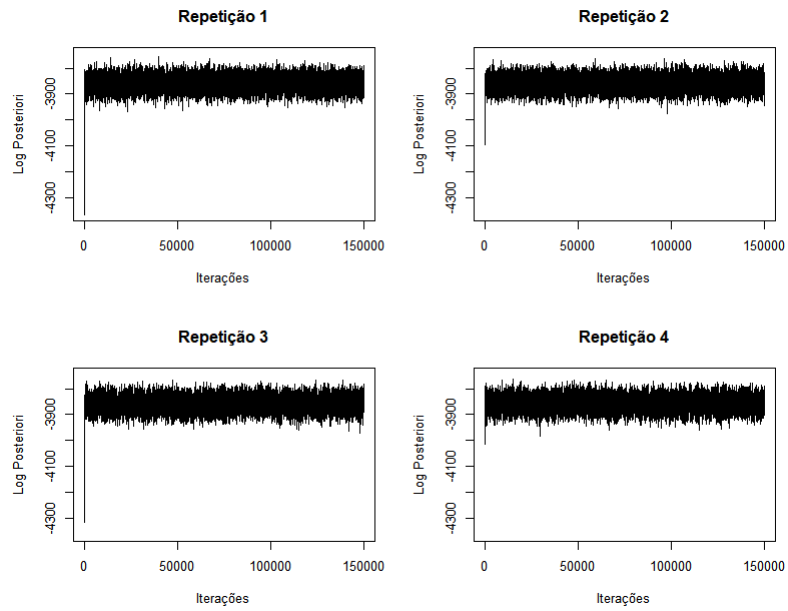


Figura 39: Cadeias Log *Posteriori* algoritmo Metropolis I para Estudo Simulado 1.

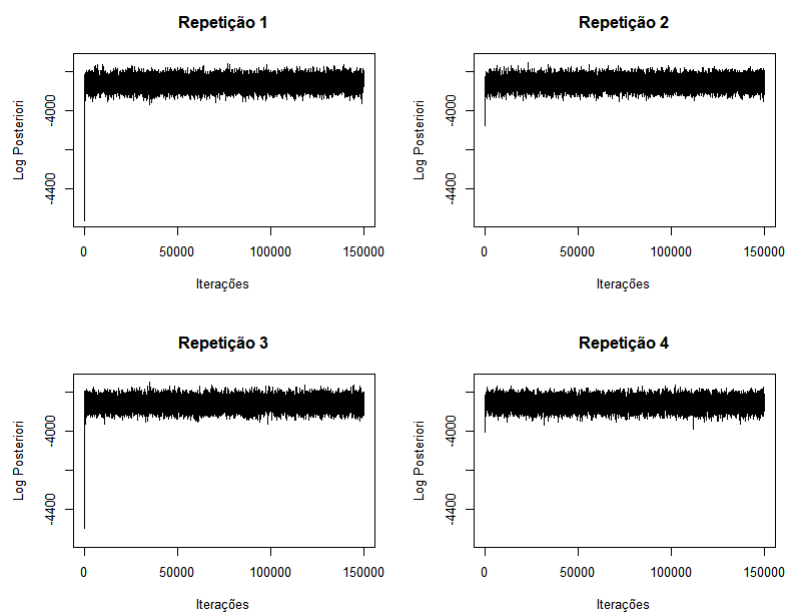


Figura 40: Cadeias *Log Posteriori* algoritmo Metropolis II para Estudo Simulado 1.

## C.2 Estudo Simulado 2

Para melhor visualização dos gráficos das cadeias da função log densidade *posteriori*, foram ignoradas as 50 primeiras iterações. Valor inicial das cadeias: -49694.37, -32761.26, -55063.43 e -31930.89.

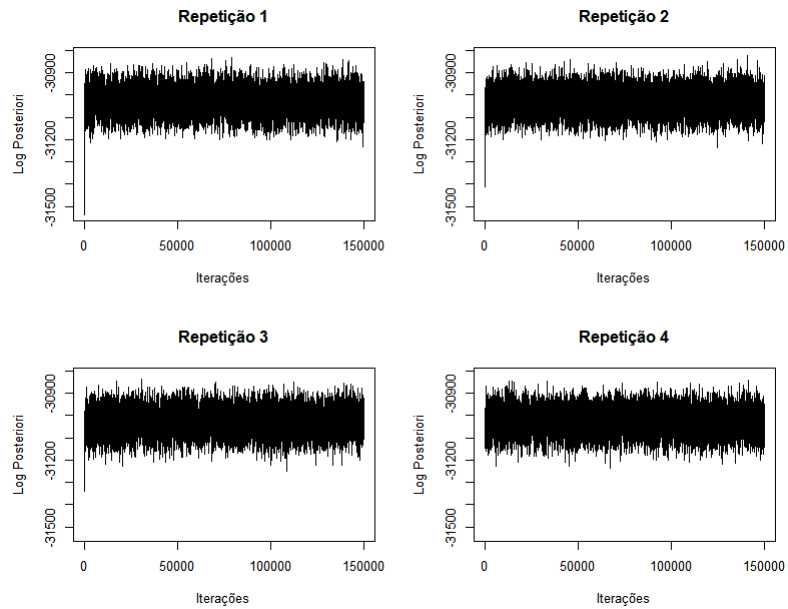


Figura 41: Cadeias Log *Posteriori* algoritmo Béguin and Glas (2001) para Estudo Simulado 2.

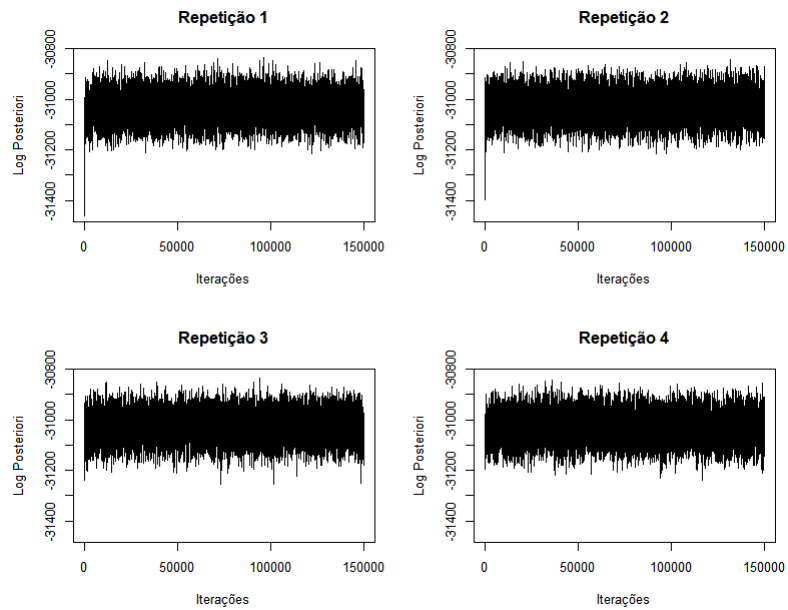


Figura 42: Cadeias Log *Posteriori* algoritmo Gonçalves et al. (2018) para Estudo Simulado 2.

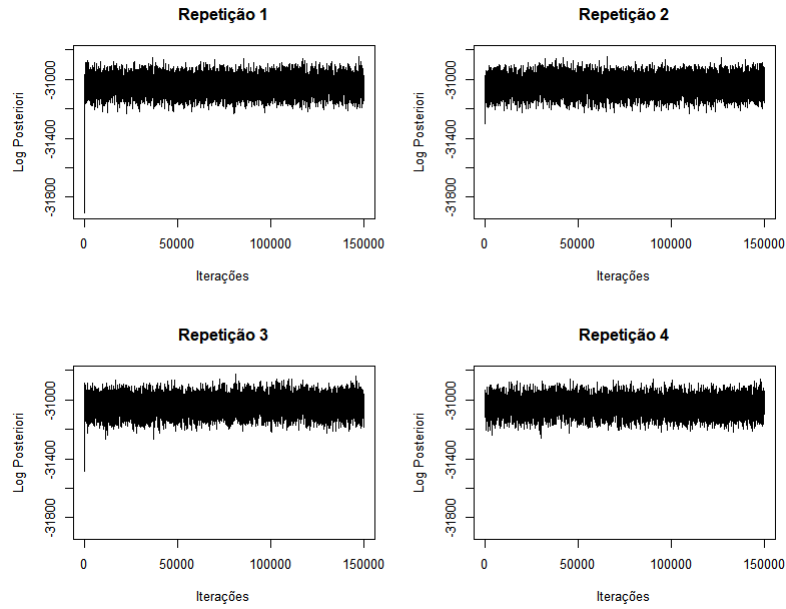


Figura 43: Cadeias *Log Posteriori* algoritmo Metropolis I para Estudo Simulado 2.

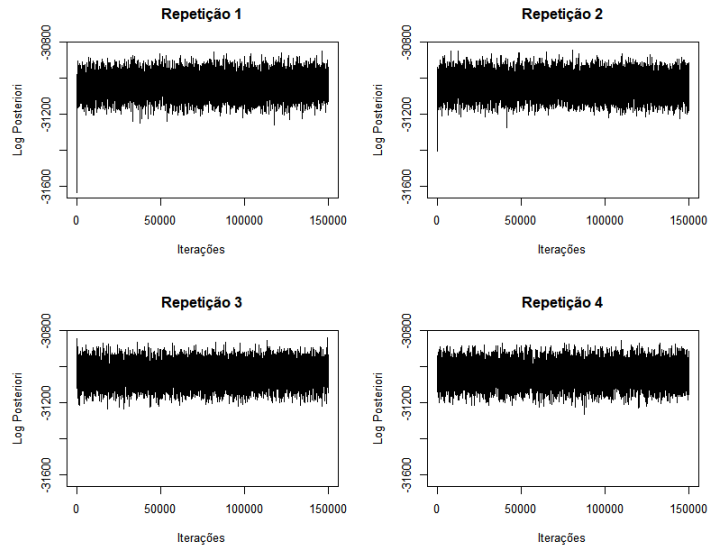


Figura 44: Cadeias *Log Posteriori* algoritmo Metropolis II para Estudo Simulado 2.

### C.3 Estudo Simulado 3

Para melhor visualização dos gráficos das cadeias da função log densidade *posteriori*, foram ignoradas as 100 primeiras iterações. Valor inicial das cadeias: -378342.5, -235234.7, -419285.3 e -227231.1.

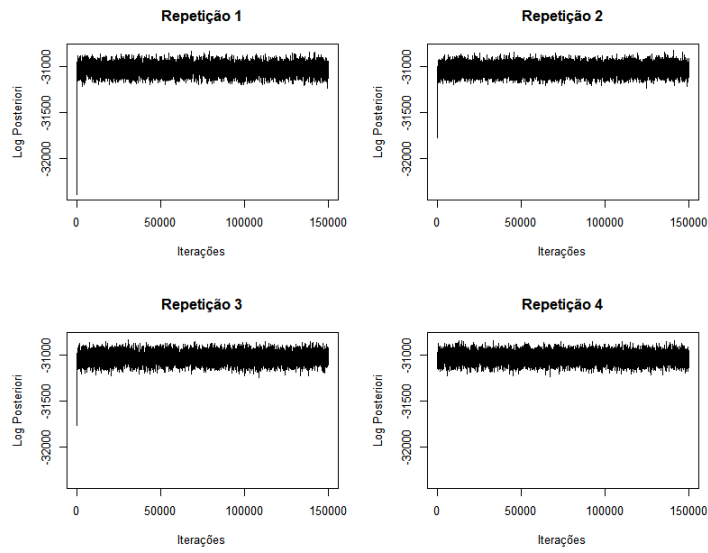


Figura 45: Cadeias Log *Posteriori* algoritmo Béguin and Glas (2001) para Estudo Simulado 3.

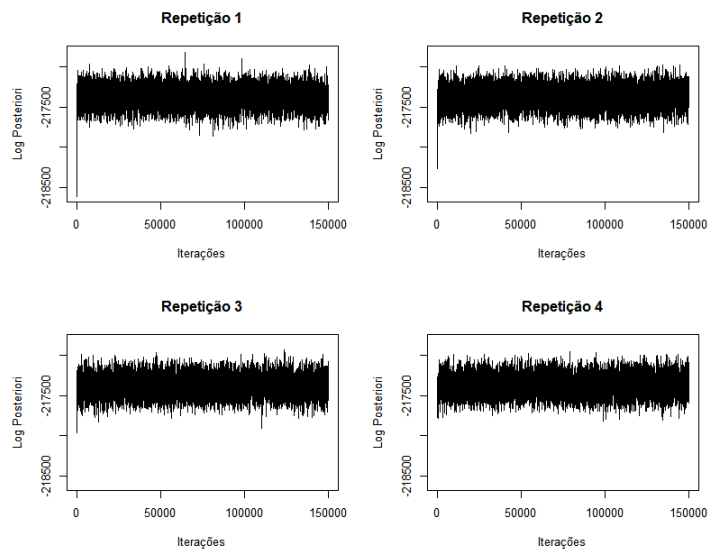


Figura 46: Cadeias Log *Posteriori* algoritmo Gonçalves et al. (2018) para Estudo Simulado 3.



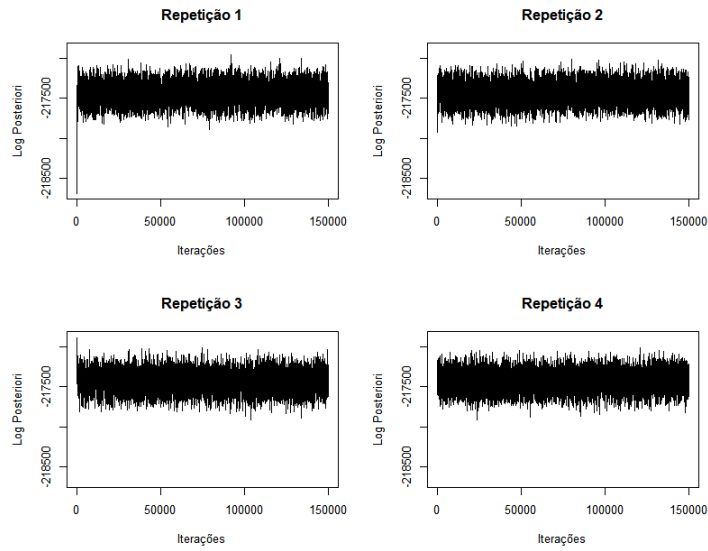


Figura 47: Cadeias Log *Posteriori* algoritmo Metropolis I para Estudo Simulado 3.

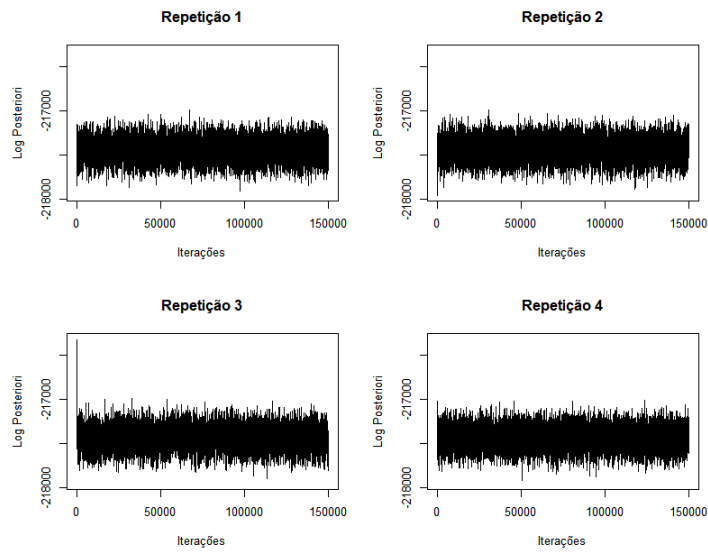


Figura 48: Cadeias Log *Posteriori* algoritmo Metropolis II para Estudo Simulado 3.