Universidade de Brasília
Instituto de Ciências Exatas
Departmento de Estatísticas

Dissertação de Mestrado

Computação Bayesiana Aproximada via fatoração da distribuição a posteriori

por

Arthur Canotilho Machado

Brasília, 16 de abril de 2023

Computação Bayesiana Aproximada via fatoração da distribuição a posteriori

por

Arthur Canotilho Machado

Dissertação apresentada ao Departamento de Estatística da Universidade de Brasília, como parte dos requisitos para obtenção do título de Mestre em Estatística.

Orientador: Prof. Dr. Guilherme Souza Rodrigues

Brasília, 16 de abril de 2023

Se eu vi mais longe, foi por estar sobre ombros de gigantes.

(Isaac Newton)

Agradecimentos

Agradeço a todas as pessoas que de alguma forma passaram pela minha vida e, certamente, têm uma parte de contribuição nessa conquista. Em particular, a Deus, a minha família, amigos, meu professor orientador, aos membros do grupo de trabalho pelo qual o tema dessa dissertação surgiu, sendo eles Débora Cristiane dos Santos, Guilherme Rodrigues, Chris Drovandi, David J. Nott, Scott Sisson, a Universidade de Brasília, professores e funcionários do Departamento, colegas do trabalho, e muitos outros.

Este estudo foi financiado em parte pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

É comum em problemas modernos de Inferência Bayesiana se deparar com dados complexos e/ou de alta dimensão, como os que surgem no campo da genética de populações (Beaumont, Zhang e Balding, 2002), para os quais a função de verossimilhança e as distribuições marginas são difíceis de serem computadas ou até mesmo intratáveis, gerando, assim, problemas na obtenção da distribuição a posteriori. Existem diversos métodos de aproximação da distribuição a posteriori para esses tipos de casos, entre eles o Amostrador de Gibbs aproximado, proposto por Rodrigues, Nott e Sisson (2019), o qual permite a geração de amostras de uma distribuição a posteriori aproximada usando princípios da Computação Bayesiana Aproximada (ABC) e do Amostrador de Gibbs. Santos (2021) propôs um aprimoramento da técnica a partir da descorrelação prévia dos parâmetros de interesse e do uso de modelos de regressão quantílica via redes neurais no processo de aproximação das distribuições condicionais completas. Neste trabalho sugerimos a substituição do Amostrador de Gibbs aproximado por um algoritmo que aproxima distribuições definidas por uma fatorações conveniente da distribuição a posteriori. São apresentadas uma revisão da teoria e aplicações práticas comparando os métodos de Rodrigues, Nott e Sisson (2019), de Santos (2021) e o proposto neste trabalho. Foram gerados conjuntos de dados sintéticos para comparação dos métodos. O algoritmo proposto neste trabalho mostrou boa performance comparado aos seus pares, apresentando um avanço na técnica.
Palavras-chave: Distribuição a posteriori. Distribuições condicionais. Regressão Quantílica via Redes Neurais. Verossimilhança intratável.

Abstract

It is common in modern Bayesian inference problems to come across complex and/or high-dimensional models, such as those that arise in the field of population genetics (Beaumont, Zhang, Balding, 2002), where the likelihood function and marginal distributions are difficult or even intractable to compute, leading to problems in obtaining the posterior distribution. There are several methods for approximating the posterior distribution for these type of cases, including the Approximate Gibbs Sampler proposed by Rodrigues, Nott, and Sisson (2019), which allows the generation of samples from an approximate posterior distribution using principles of Approximate Bayesian Computation (ABC) and Gibbs Sampling. Santos (2021) proposed an improvement to the technique by previously decorrelating the parameters of interest and using quantile regression models via neural networks in the process of approximating the complete conditional distributions. In this work, we suggest replacing the Approximate Gibbs Sampler with an algorithm that approximates the terms of a convenient factorization of the posterior distribution. We present a review of the theory and practical applications comparing the methods of Rodrigues, Nott, and Sisson (2019), of Santos (2021), and the proposed in this work. Synthetic datasets were generated to compare the methods. The algorithm proposed in this work showed good performance compared to its peers.

Key-words: Posteriori distribution. Conditional distributions. Quantile Regression via Neural Networks. Intractable likelihood. Likelihood-free methods.

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

O estudo das distribuições de densidades condicionais permite a quantificação da incerteza sobre um determinado conjunto de parâmetros de um modelo probabilistico. Diversas abordagens foram desenvolvidas para estimar valores usando dados observados, incluindo a Bayesiana, na qual os parâmetros do modelo são tratados como variáveis aleatórias. A inferência Bayesiana tem como base o Teorema de Bayes que permite combinar a informação de conhecimentos prévios (distribuição a priori) com a informação advinda dos dados (função de verossimilhança), resultando, assim, na distribuição a posteriori. Dessa forma, a distribuição a posteriori reflete todo o conhecimento atualizado sobre o parâmetro e pode ser denotada da seguinte forma:

$$\pi(\boldsymbol{\theta}|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{p(\boldsymbol{X})},$$

onde $\pi(\boldsymbol{\theta})$ representa a distribuição a priori do vetor de parâmetros $\boldsymbol{\theta}$, $p(\boldsymbol{X}|\boldsymbol{\theta})$ a função de verossimilhança de um conjunto de dados $\boldsymbol{X}$ e $p(\boldsymbol{X})$ a função de verossimilhança marginal dos dados, que atua como uma constante de normalização para garantir que a distribuição a posteriori seja uma distribuição de probabilidade válida.

A partir da distribuição a posteriori, é possível modelar a incerteza de forma mate-

mática, através da utilização de probabilidades. Ela contém toda a informação necessária para análise do modelo, incluindo checagem e validação do modelo, predição de valores e tomada de decisões. Sua inferência ocorre sobre a distribuição dos parâmetros obtidos, não necessitando, assim, de convergência assintótica dos parâmetros do modelo, como ocorre na abordagem frequentista.

Um dos grandes problemas da inferência Bayesiana ocorre quando a distribuição a posteriori não tem uma forma fechada, não podendo, portanto, ser calculada analiticamente. Esse problema é devido, em geral, à complexidade dos dados que tornam sua integração de difícil cômputo, em especial em modelos de alta dimensão.

A maioria das soluções que envolvem esse tipo de problema estão relacionadas a métodos numéricos de computação, em especial os métodos de Monte Carlo via Cadeia de Markov (MCMC) e de Monte Carlo Sequencial (SMC) (Del Moral, 1996; Kinas e Andrade, 2010). Eles buscam aproximar valores de interesse a partir de simulações aleatórias de uma distribuição de interesse, realizando, assim, um esquema iterativo de simulação na qual cada iteração do algoritmo depende apenas da iteração anterior. Quando executados um número suficientemente grande de iterações a cadeia se aproximará da distribuição de equilíbrio. Por isso, não é necessária a utilização de soluções exatas ou equações diferenciais para seu cômputo.

Outro problema muito comum nesse tipo de abordagem é a intratabilidade da função de verossimilhança. Em geral, resulta da complexidade das relações probabilísticas existentes, como ocorre em estudos no campo da genética.

Com isso, não é possível obter amostras da distribuição a posteriori, já que a função de verossimilhança é parte fundamental da modelagem Bayesiana e nos métodos MCMC. Uma alternativa é a construção de um modelo diferente mas que se aproxima da distribuição original. Esses métodos são chamados *likelihood-free*, pois não exigem o calculo direto da função de verossimilhança.

Em especial, e foco deste trabalho, está o método de Computação Bayesiana Apro-

ximada (ABC, em inglês *Approximate Bayesian Computation*) que tem se tornado um método eficiente e de bons resultados. O método utiliza como base o algoritmo de amostragem por rejeição, em que são geradas amostras de uma distribuição, e essas são estão aceitas com uma certa probabilidade.

Pela sua forma de aproximação da distribuição a posteriori, este método tem ganhado destaque por não precisar de uma expressão analítica para a função de verossimilhança e nem realizar cômputos sobre a distribuição marginal $p(\boldsymbol{X})$. Por isso, é considerado um método *likelihood-free*.

Beaumont, Zhang e Balding (2002) cunharam o termo ABC em seu estudo sobre populações na área de genética. A principal contribuição desse artigo foi introduzir a ideia de aproximação da distribuição por meio de estatísticas-resumo dos dados. A condição de aproximar uma distribuição a outra é algo muito custoso computacionalmente e, por vezes, impossível. Com isso, reduz-se os dados em estatísticas-resumo, assim, uma distribuição se aproxima da outra quando suas estatísticas-resumo são próximas entre si. Valendo da seguinte premissa:

$$p(\boldsymbol{\theta}|\boldsymbol{X_{obs}}) \approx p(\boldsymbol{\theta}|\boldsymbol{S}(\boldsymbol{X_{obs}})),$$

na qual $\boldsymbol{\theta}$ é o vetor de parâmetros da distribuição, $\boldsymbol{X_{obs}}$ os dados observados e $\boldsymbol{S}(\boldsymbol{X_{obs}})$ as estatísticas-resumo observadas.

Com essa alteração, substitui-se a abordagem que trabalha com os dados completos por uma abordagem que se restringe a poucos parâmetros, o que diminui consideravelmente o custo computacional.

Segundo Rodrigues, Nott e Sisson (2019), os métodos baseados no algoritmo ABC têm bom desempenho em problemas com baixa ou moderada dimensão (quantidade de parâmetros menor do que 50).

Grandes problemas ainda enfrentados pelo ABC estão na especificação da distribuição

a priori, escolha da medida de distância que será utilizada para aproximar as distribuições e na escolha das estatísticas-resumo.

A escolha da estatística resumo é avaliada em diversos estudos, seja quanto a sua suficiência, ou quanto a sua dimensionalidade, o chamado *curse-of-dimensionality* (Blum e Francois, 2010 e Nott et al., 2018b), problema encontrado quando o algoritmo tem dificuldade de gerar amostras aproximadas da distribuição a posteriori devido à baixa probabilidade de aceitação das amostras candidatas. Para evitar esse problema, uma alternativa é considerar estatísticas-resumo com dimensão menor ou igual a dimensão dos dados observados.

Esse problema diz que mais estatísticas-resumo não necessariamente promovem em uma melhor aproximação. Logo, a escolha delas deve ser de forma racional e, com isso, de difícil obtenção, não havendo uma solução comum para todos os modelos (Sisson, Fan e Beaumont, 2018). Um exemplo é o caso da distribuição Poisson que tem como estatísticas-resumo de média e variância iguais. Em geral, a melhor escolha para uma estatística resumo é a estatísica mínima suficiente, sendo que esta é, muita vezes, difícil de obter quanto mais os dados se tornam complexos. A noção de suficiência acaba sendo absorvida pelo erro de aproximação. Porém, é de se deixar claro, que quanto pior a estatísticas-resumo pior será a aproximação.

Beaumont, Zhang e Balding (2002) propuseram realizar a aproximação do algoritmo ABC através do ajuste de uma regressão linear, usando um modelo linear local para minimizar o erro na aproximação dos parâmetros simulados e as estatísticas-resumo geradas. Modelos não lineares foram depois introduzidos para obter melhores aproximações da distribuição original, como o uso de modelos hierárquicos (Bazin, Dawson e Beaumont, 2010) e modelos de regressão heterocedástica condicional não linear (Blum e Francois, 2010).

Com os resultados dos modelos de regressão, Nott et al. (2014) apresentaram a ideia de reconstruir a distribuição posteriori conjunta com base nas distribuições marginais a posteriori obtidas, chamado de ajuste marginal.

Uma técnica para contornar o problema da dimensionalidade e melhorar a eficiência foi apresentada em Rodrigues, Nott e Sisson (2019) que combina o uso do Amostrador de Gibbs com o ABC, com o intuito de substituir as distribuições condicionais completas (DCC) por aproximações obtidas por modelos de regressão dentro do algoritmo de Gibbs, não necessitando, assim, do cálculo da função de verossimilhança. Além disso, o uso do Amostrador de Gibbs simplifica a estrutura dos modelos de regressão, lidando, assim, com modelos de baixa dimensão ao invés de regressões multivariadas obtidas pela técnica de ajuste de regressão.

Mas, pelo fato de ainda se utilizar do Amostrador de Gibbs, foram verificados, segundo Rodrigues, Nott e Sisson (2019), problemas na velocidade de mistura da cadeia, o que impacta na convergência dos parâmetros. Esse tipo de problema é característico do algoritmo de Gibbs e que leva a uma baixa eficiência do mesmo.

Santos (2021) aprimora o algoritmo de Rodrigues, Nott e Sisson (2019), com implementando o uso do modelo de regressão quantílica via redes neurais juntamente com a interpolação monotônica Splines para obter as densidades condicionais completas e assim utilizá-las no Amostrador de Gibbs. Além da utilização da descorrelação prévia dos parâmetros para acelerar a convergência e aumentar a velocidade de mistura.

Os estudos de simulação realizados em Santos (2021) para os dados de uma distribuição Normal Bivariada e uma distribuição de Mistura de Normais mostraram que a substituição de modelos menos flexíveis, de média e variância, para modelos de quantis obteve melhores aproximações para as DCCs, reduzindo, assim, o erro da obtenção da distribuição a posteriori de interesse. O tempo computacional com as novas implementações não se mostrou diferente, mas a velocidade de mistura da cadeia de Markov se mostrou drasticamente maior, na qual foi observada a eliminação da autocorrelação na cadeia. Porém, ainda são herdadas as características e problemas do Gibbs.

O fato da velocidade de mistura da cadeia de Markov ser maior influencia no tempo computacional, pois são necessárias menos iterações para a convergência para o verdadeiro

parâmetro. Porém, como foram geradas as mesmas quantidades de iterações para os métodos, então não houve ganho computacional. Mas, se utilizássemos uma regra de parada para a convergência, o tempo computacional certamente seria menor.

O ganho computacional aqui estudado está em avaliar se gerar amostras via Amostrador de Gibbs ou via fatorações da distribuição a posteriori impactam no custo computacional. Ou até mesmo se mais operações, como o uso da descorrelação prévia dos parâmetros, aumentam o tempo computacional.

Outra contribuição do estudo de Santos (2021) é de que, ao invés de fazer uma expansão no espaço paramétrico e deixar mais simples as regressões condicionais completas, a solução foi estimar diretamente no espaço original (desde que o modelo utilizado seja mais sofisticado).

O ajuste do modelos não obteve, muita das vezes, bons resultados, e isso pode ser um fato causado pelo uso do modelo de regressão quantílica e o uso das redes neurais, que tendem a apresentar melhor desempenho quando são treinadas sobre grandes volumes de dados.

Vale mencionar que o uso da descorrelação dos parâmetros resultou em piora do modelo para o exemplo da Mistura de Normais, mostrando que ela deve ser feita quando a mistura da cadeia se mostrar lenta e os parâmetros forem altamente correlacionados na distribuição a posteriori. Pelo fato do modelo ser mais flexível, a distribuição a posteriori consegue percorrer mais o espaço, apresentando, assim, melhores resultados do que o métodos de Rodrigues, Nott e Sisson (2019).

Observou-se, a partir dos resultados de Santos (2021), que quando se tem um modelo flexível suficiente para obter as DCCs, elas deixam de ser necessárias, com isso, pode-se criar uma amostra independente dos seus resultados fazendo uma fatoração da distribuição a posteriori.

Essa técnica combinada com os métodos ABC tem se mostrado eficiente para a função de verossimilhança em altas dimensões. Bazin et al. (2010), Barthelmé and Chopin (2014)

e White et al. (2015) realizaram estudos dessa ténica junto com modelos hierárquicos, esquemas de *expectation-propagation* e cadeias de Markov. O fato de se usar a fatoração na função de verossimilhança implica em componentes de baixa dimensão, facilitando na comparação de estatísticas-resumo, que agora estarão, também, em baixa dimensão e evitando o problema de *curse-of-dimensionality*.

A presente dissertação consiste em um estudo comparativo dos métodos ABC abordados em Rodrigues (2019) e Santos (2021), além da proposição de um novo método que junta os anteriores já citados, foi-se adicionado que a obtenção dos parâmetros se dará por uma partição da distribuição a posteriori e, assim, investigar comparativamente suas performances. Este último método é baseado nas propostas futuras de abordagens de Rodrigues (2019) e estima as densidades condicionais via ABC e regressão quantílica por redes neurais com o uso da correção Splines Monotônico e gera a distribuição a posteriori a partir de fatorações das DCCs, evitando, assim, o uso do algoritmo de Gibbs.

O ganho deste método está na independência das amostras geradas, e, com isso, o ganho em tempo computacional é considerável, dado que não é necessário realizar diversas atualizações nos parâmetros para obter a melhor aproximação. Além disso, não é necessário o uso da descorrelação prévia dos parâmetros já que eles são independentes, portanto, autocorrelação zero entre as amostras.

Dados sintéticos foram gerados a partir de diferentes cenários de exemplos (distribuição Normal Bivariada, distribuição *Twisted Gaussian* e distribuição Mistura de Normais) e avaliados quanto a velocidade de convergência, autocorrelação dos parâmetros e distribuição marginal e conjunta geradas.

Este trabalho está organizado da seguinte forma: Seção 2 realiza uma revisão bibliográfica das técnicas utilizadas nos estudos de simulação, sendo elas o método ABC, o Amostrador de Gibbs, a descorrelação dos parâmetros, uso do modelo de regressão quantílica via redes neurais com correção Splines Monotônico e, por fim, os métodos propostos por Rodrigues, Nott e Sisson (2019) e Santos (2021). Já, a Seção 3 descreve o método

proposto como inovação para este trabalho, chamado de Aproximação via fatorações da distribuição a posteriori. A Seção 4 apresenta os estudos simulados, com os cenários gerados e a comparação dos resultados entre as diferentes implementações. Ao final, tem-se a Seção 5 com a conclusão do estudo apresentando resumidamente os resultados e apontamentos para trabalhos futuros.

Todas as implementações, incluindo as de redes neurais profundas feitas por meio dos pacotes Keras e Tensorflow, foram realizadas pelo software R por meio da plataforma Google Colab Pro que fornece a possibilidade de uso de GPUs e TPUs, além de uma memória RAM mais elevada do que os computadores convecionais.

Os códigos utilizados nesta dissertação podem ser encontrados no Github (`https://github.com/artemenseken/ABC-factorization-posteriori-R/tree/main`).

Capítulo 2

Revisão bibliográfica

Neste capítulo serão apresentados os principais métodos utilizados nos algoritmos, bem como a descrição dos algoritmos de Rodrigues, Nott e Sisson (2019) e de Santos (2021).

2.1 Métodos ABC

Por meio da abordagem Bayesiana, tem-se o propósito de obter a distribuição a posteriori a partir da fórmula

$$\pi(\boldsymbol{\theta}|\boldsymbol{X}) \propto L(\boldsymbol{X}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}),$$

sendo, $L(\boldsymbol{X}|\boldsymbol{\theta})$ a função de verossimilhança dos dados e $\pi(\boldsymbol{\theta})$ a distribuição a priori.

Quando a função de verossimilhança é computacionalmente intratável ou até mesmo impossível de ser calculada, métodos como o ABC se fazem necessários. Esses, por não lidarem diretamente com a função de verossimilhança, têm boa perfomance para este tipo de problema, e são assim chamados de *likelihood-free*.

O mecanismo simples de um algoritmo ABC, proposto por Rubin (1984) e depois melhorada por Tavare (1997), utiliza a ideia do algoritmo de rejeição. O mecanismo se baseia na geração de amostras sintéticas na avaliação da proximidade dessas amostras às estatísticas-resumo observadas. Se elas são próximas, a um nível de tolerância espe-

cífico e uma medida de distância, então se aceita as respectivas amostras candidatas, se não, rejeita-se. Uma função de pesos, $K_h(d)$, pode ser aplicada para ponderar as distâncias calculadas, sendo as maiores com menor peso. Este novo algoritmo é chamado de Amostragem por importância com a implementação da função de pesos.

Porém, aproximar uma distribuição a outra, $\boldsymbol{X}_{obs} \approx \boldsymbol{X}$, é muito improvável e computacionalmente muito oneroso, sendo $\boldsymbol{X}_{obs}$ a distribuição dos dados observada e $\boldsymbol{X}$ uma distribuição conhecida e tratável analiticamente. Uma alternativa é reduzir os dados a estatísticas-resumo, $s_{\boldsymbol{X}} = S(\boldsymbol{X})$. Logo, ao invés de aproximar os dados completos, aproxima-se suas estatísticas-resumo, $S(\boldsymbol{X}_{obs}) \approx S(\boldsymbol{X})$.

Gera-se, então, aproximações da distribuição a posteriori do tipo:

$$\pi_{ABC}(\boldsymbol{\theta}|S(\boldsymbol{X}_{obs})) = \int K_h(||S(\boldsymbol{X}) - S(\boldsymbol{X}_{obs})||)p(\boldsymbol{X}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\,d\boldsymbol{X}$$

Os passos do algoritmo são:

i Geram-se amostras de uma distribuição sintética $\boldsymbol{X}$ e com parâmetro $\boldsymbol{\theta}$, dado por $(\theta_1, X_1), (\theta_2, X_2), ..., (\theta_N, X_N)$, através da fórmula

$$\pi(\boldsymbol{\theta}, \boldsymbol{X}) = L(\boldsymbol{X}|\boldsymbol{\theta})\pi(\boldsymbol{\theta});$$

ii Computa-se as estatísticas-resumo $s_{\boldsymbol{X}} = S(\boldsymbol{X})$;

iii Se $s_{\boldsymbol{X}}$ é próximo suficiente, a partir de uma métrica de distância, de $s_{\boldsymbol{X}_{obs}}$ então: $\pi(\boldsymbol{\theta}|s_{\boldsymbol{X}} \approx s_{\boldsymbol{X}_{obs}}) \approx \pi(\boldsymbol{\theta}|s_{\boldsymbol{X}_{obs}}) \approx \pi(\boldsymbol{\theta}|\boldsymbol{X}_{obs})$.

O algoritmo de rejeição é utilizado na verificação da aproximação, seu objetivo é de aceitar o parâmetro $\boldsymbol{\theta}$ com probabilidade $\propto \frac{f(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}$, em que $\boldsymbol{\theta} \sim g(\boldsymbol{\theta})$.

A equação $\frac{f(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}$ pode ser reescrita como:

$$\frac{f(\boldsymbol{\theta})}{g(\boldsymbol{\theta})} \propto \frac{\pi(\boldsymbol{\theta}, s_{\boldsymbol{X}}|s_{\boldsymbol{X}_{obs}})}{L(s_{\boldsymbol{X}}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})} \propto \frac{K_h(||s_{\boldsymbol{X}} - s_{\boldsymbol{X}_{obs}}||)L(s_{\boldsymbol{X}}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{L(s_{\boldsymbol{X}}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})} = K_h(||s_{\boldsymbol{X}} - s_{\boldsymbol{X}_{obs}}||)$$

Onde,

$$K_h(||s_{\boldsymbol{X}} - s_{\boldsymbol{X}_{obs}}||) = \begin{cases} 1, & \text{se } ||s_{\boldsymbol{X}} - s_{\boldsymbol{X}_{obs}}|| \leq h \\ 0, & \text{caso contrário} \end{cases} \tag{2.1}$$

Logo, pode-se interpretar $\pi_{ABC}(\boldsymbol{\theta}|s_{\boldsymbol{X}_{obs}}) = \pi(\boldsymbol{\theta}|s_{\boldsymbol{X}_{obs}})$, sendo baseado na estimação de argumentos pela densidade Kernel.

Tem-se como resultado final do algoritmo um conjunto de parâmetros $\theta^{(1)}, ..., \theta^{(N)}$ que são amostras de $\pi(\boldsymbol{\theta}|s_{\boldsymbol{X}_{obs}})$.

A estrutura de um algoritmo de Amostragem por importância é dado abaixo.

Algoritmo 1: Amostragem por importância ABC

Entrada:

Um conjunto de dados observados ($\boldsymbol{X_{obs}}$);
Uma distribuição a priori $\pi(\boldsymbol{\theta})$;
Um procedimento para gerar dados sob o modelo $p(\boldsymbol{X}_{obs}|\boldsymbol{\theta})$;
Um inteiro positivo $N > 0$;
Um vetor de medidas resumo observadas $s_{\boldsymbol{X}_{obs}} = S(\boldsymbol{X}_{obs})$;
Uma função kernel $K_h(u)$ e um parâmetro de escala $h > 0$;

// Início

para $i = 1, 2, \cdots, N$ **faça**

1.1 Gere $\boldsymbol{\theta}^{(i)} \sim \pi(\boldsymbol{\theta})$ para a distribuição a priori;
1.2 Gere $\boldsymbol{X}^{(i)} \sim p(\boldsymbol{X}|\boldsymbol{\theta}^{(i)})$ para a função de verossimilhança;
1.3 Calcule as medidas resumo $s^{(i)} = S(\boldsymbol{X}^{(i)})$;
1.4 Atribua $\boldsymbol{\theta^{(i)}}$ o peso $w_i \propto K_h(||s^{(i)}_{\boldsymbol{X}} - s_{\boldsymbol{X}_{obs}}||)$;

fim

Saída:

Um conjunto de vetores de parâmetros ponderados $\{\boldsymbol{\theta}, w\}_{i=1}^{N} \sim \pi_{ABC}(\boldsymbol{\theta}|s_{\boldsymbol{X}_{obs}})$.

2.1.1 Ajuste do ABC por regressão

Beaumont, Zhang e Balding (2002) propuseram o ajuste do ABC por modelos de regressão baseando-se na modelagem coalescente de populações em genética. Depois do passo de aceitação dos parâmetros, via algoritmo de rejeição, estes são ajustados para considerar as diferenças entre as estatísticas-resumo observadas e simuladas.

Este método é bastante usado em implementações do ABC, com grandes resultados na melhora da acurácia do modelo. O pós-processamento do ABC, segundo Blum e Francois (2010) melhora na qualidade da aproximação e na eficiência computacional do algoritmo.

Beaumont, Zhang e Balding (2002) introduziram o ajuste usando um modelo linear local na vizinhança de $s_{\boldsymbol{X}_{obs}}$:

$$\theta^{(i)} = m(s^{(i)}_{\boldsymbol{X}}) + \epsilon^{(i)}, \text{em que } i = 1, ..., N,$$

na qual $m(s^{(i)}_{\boldsymbol{X}})$ é a esperança condicional de $\theta|s_{\boldsymbol{X}}$ e $\epsilon^{(i)}$ o erro. Este modelo assume

homocedasticidade, tendo a variância dos resíduos não dependendo de $s_{\boldsymbol{X}}$.

Logo, a estimação de θ por meio do modelo se dá por:

$$\theta^{(i)} = \alpha_d + \beta_d^T(s_{\boldsymbol{X}}^{(i)} - s_{\boldsymbol{X}_{obs}}) + \epsilon_d^{(i)},$$

sendo $i = 1, ..., N$, d a dimensão do vetor de parâmetros, α_d e β_d os coeficientes da regressão e $\epsilon_d^{(i)}$ os resíduos do modelo, que seguem uma $N(0, \sigma_d^2)$.

Ao se gerar amostras ponderadas, garante-se que seja dada maior importância às amostras mais próximas aos dados observados s_{obs}.

Outros modelos de regressão foram propostos para melhorar as estimativas, como modelos que considerem a heterocedasticidade dos dados (Blum e Francois, 2010) com uso de redes neurais para estimar as funções de média e variância condicional não linear, ou com modelos mais robustos como a regressão (Ridge, Blum et al., 2013), e regressão quantílica via redes neurais (Santos, 2021).

2.2 Amostrador de Gibbs

O Amostrador de Gibbs é um algoritmo pertencente a família de algoritmos MCMC (Monte Carlo via Cadeias de Markov). Estes são utilizados para obter numericamente uma aproximação da distribuição a posteriori quando essas não apresentam o seu núcleo de forma conhecida, problema frequentemente enfrentado em Inferência Bayesiana.

Seus métodos simulam de uma densidade $p(\boldsymbol{\theta})$ de interesse através da produção de uma cadeia de Markov homogênea, ergódica e irredutível sendo a distribuição de $p(\boldsymbol{\theta})$ estacionária.

O Amostrador de Gibbs funciona da seguinte forma: seja θ um vetor de parâmetros, $p(\boldsymbol{\theta})$ a densidade conjunta de $\boldsymbol{\theta}$ e $p(\theta_i|\theta_{-i})$ a distribuição condicional completa dos θ_i dados todos os outros parâmetros $\theta_{-i} = (\theta_1, ..., \theta_{i-1}, \theta_{i+1}, ..., \theta_p)$, então o algoritmo gerará uma sequência $\theta^{(0)}, \theta^{(1)}, ..\theta^{(N)}$. a partir de uma cadeia de Markov cuja a distribuição de equilí-

brio é $p(\boldsymbol{\theta})$ e o núcleo da distribuição é dado pelo produto das distribuições condicionais completas.

Os passos do algoritmo são:

i) Assumir valores iniciais arbitrários para os $\theta^{(j)}$.

ii) Na j-ésima iteração, sortear um $\theta^{(j)}$ a partir de $\theta^{(j-1)}$ de modo que:

a) Gere $\theta_1^{(j)}$ de $p(\theta_1|\theta_2^{(j-1)}, ..., \theta_p^{(j-1)})$,

b) Gere $\theta_2^{(j)}$ de $p(\theta_2|\theta_1^{(j)}, \theta_3^{(j-1)}, ..., \theta_p^{(j-1)})$,

c) Gere $\theta_3^{(j)}$ de $p(\theta_3|\theta_1^{(j)}, \theta_2^{(j)}, \theta_4^{(j-1)}, ..., \theta_p^{(j-1)})$, etc.

d) E finalmente, gere $\theta_p^{(j)}$ de $p(\theta_p|\theta_1^{(j)}, \theta_2^{(j)}, \theta_3^{(j)}, ..., \theta_{p-1}^{(j)})$

Com isso, é construído o vetor $\theta_1^{(j)} = (\theta_1^{(j)}, ..., \theta_p^{(j)})$ e, sob certas condições de regularidade, quando $j \to \infty$ a distribuição limite de $\theta^{(j)}$ é $p(\theta)$.

Ao construir uma Cadeia de Markov que tenha distribuição estacionária pode-se gerar uma quantidade suficientemente grande que implica em uma convergência da cadeia para a distribuição alvo.

Algumas características negativas desse método são observadas: mistura possivelmente lenta da cadeia, causada pela autocorrelação das amostras e problemas de convergência.

2.3 Descorrelação dos parâmetros

A descorrelação dos parâmetros é uma técnica matemática empregada para tornar a covariância entre eles igual a zero, fazendo com que o valor esperado do produto seja igual ao produto dos valores esperados, $E(XY) = E(X)E(Y)$.

Quando a covariância é zero, tem-se que a correlação linear entre as variáveis, para o caso, os parâmetros, também é zero. Isso se reflete na velocidade de convergência da distribuição alvo da cadeia de Markov, que, segundo Paulino, Turkman e Murteira (2003), depende do grau de correlação entre os elementos do vetor aleatório. Segundo o artigo, o Amostrador de Gibbs tende a se movimentar mais pelas distribuições condicionais

completas quando suas componentes são independentes ou fracamente correlacionadas.

A descorrelação engloba uma série de métodos estatísticos que visam satisfazer a proposição matemática citada no parágrafo anterior. O principal método utilizado para isso e usado neste estudo é o de Análise de Componentes Principais (PCA). A PCA é muito empregada como forma de redução da dimensionalidade em uma base de dados, para estimação de fatores na análise fatorial e a eliminação de multicolinearidade em uma regressão.

Ela consiste em obter componentes que sejam ortogonais entre si, descorrelacionadas, por meio de combinações lineares que representem a informação que foi possível abstrair das variáveis originais.

O PCA funciona da seguinte forma: seja $\boldsymbol{X^T} = [X_1, ..., X_p]$ o vetor aleatório com matriz de covariância associada $\boldsymbol{\Sigma}$. Se as combinações lineares de $\boldsymbol{X^T}$ forem definidas como

$$Y_i = \boldsymbol{a_i^T X} = a_{i1}X_1 + a_{i2}X_2 + ... + a_{ip}X_p, i = 1, 2, ..., p,$$

então, tem-se que

$$Var(Y_i) = \boldsymbol{a_i^T \Sigma a_i}, i = 1, 2, ..., p$$

$$Cov(Y_i, Y_k) = \boldsymbol{a_i^T \Sigma a_k}, i, k = 1, ..., pk$$

Com as devidas combinações lineares, sendo elas ortogonais, e considerando que as respectivas variâncias sejam as máximas possíveis, tem-se as componentes principais não correlacionadas, podendo ser escrita como:

$$Y_i = \boldsymbol{e_i^T X} = e_{i1}X_1 + e_{i2}X_2 + ... + e_{ip}X_p,$$

na qual e_{ip} é o i-ésimo autovetor gerados. Com isso, obtém-se, os seguintes resultados.

$$Var(Y_i) = \boldsymbol{e_i^T \Sigma e_i} = \lambda_i, i = 1, 2, ..., p$$

$$Cov(Y_i, Y_k) = \boldsymbol{e_i^T \Sigma e_k} = 0, i \neq k.$$

Tem-se, portanto, que as componentes principais são não correlacionadas e têm variâncias iguais aos autovalores.

Para este estudo, foi-se utilizado uma adaptação do PCA convencional, chamado de PCA ponderado. Este consiste na adoção de pesos para amostras menos ruidosas, isso faz com que os dados não sejam tratados de forma uniforme, fazendo com que se explore mais a conjuntura das amostras.

2.4 Regressão quantílica via Redes Neurais e Splines Monotônico

O uso dos modelos de regressão dentro dos algoritmos ABC está ligado aos procedimentos de ajuste ABC por regressão, introduzido por Beaumont, Zhang e Balding (2002), que permitem aproximar distribuições de probabilidade usando modelos de regressão ajustados sobre dados sintéticos. Com isso, é possível reconstruir a distribuição a posteriori conjunta aproximando as densidades condicionais.

A escolha do modelo quantílico para a regressão se dá pelo fato de ser um modelo bastante flexível e que estima os quantis da distribuição, podendo se adequar com bastante facilidade a diferentes tipos de dados. Logo, ao invés de modelar apenas as médias e variâncias, obtém-se melhores aproximações para as Densidades Condicionais Completas (DCC's), reduzindo o erro na obtenção da distribuição à posteriori de interesse.

Por ser mais flexível, esse tipo de modelo de regressão tem bastante aplicação em dados mais complexos, como distribuições que apresentam heteroscedasticidade, assimetria ou caudas pesadas.

Esse modelo de regressão foi introduzido por Koenker e Bassett (1978) e tem sido amplamente estudado desde então. A técnica é baseada na minimização da soma ponderada dos desvios absolutos entre os valores observados e preditos da variável dependente em cada quantil. Os quantis são estimados através de algoritmos iterativos, como a descida do gradiente.

Uma das vantagens da regressão quantílica é que ela permite que a relação entre as variáveis independentes e a variável dependente varie em diferentes partes da distribuição da variável dependente. Isso é útil em situações em que a relação entre as variáveis não é constante em toda a distribuição da variável dependente. Essa técnica se mostra como alternativa para a regressão local, em que é realizada uma regressão e desta são obtidos os quantis da distribuição, sendo o local cada quantil. A partir dessa regressão é possível construir a distribuição acumulada da variável aleatória..

Logo, o q-ésimo quantil é calculado a partir de β_q, que são os coeficientes de regressão para cada quantil, e predito da seguinte forma:

$$P(y \leq \boldsymbol{x_i'}\beta_q|x) = q$$

Para obter as estimativas β_q é necessário realizar um processo de minimização da função de perda dada abaixo:

$$\mathcal{L}_n(\beta_q|\boldsymbol{y}, \boldsymbol{X}) = \sum_{i:e_{i,q}\geq 0}^{n} q|y_i - \boldsymbol{x_i'}\boldsymbol{\beta_q}| + \sum_{i:e_{i,q}<0}^{n} (1-q)|y_i - \boldsymbol{x_i'}\boldsymbol{\beta_q}|,$$

sendo $e_{i,q} = y_i - \boldsymbol{x_i'}\boldsymbol{\beta_q}$, e cada β_q sendo o efeito marginal das variáveis explicativas $\boldsymbol{X}$ no q-ésimo quantil de Y.

A função de perda mais usada em problemas de regressão é a função de erro quadrático médio. Quando exponencializada ao negativo dessa função, o resultado gerado é a distribuição gaussiana, sendo a moda correspondente ao parâmetro de média μ nesta

distribuição. A perda para uma observação individual é dada por:

$$\mathcal{L}(e_i|q) = \begin{cases} qe_i,\ e_i \geq 0, \\ (1-q)e_i,\ e_i < 0 \end{cases},$$

na qual q é o quantil que se quer estimar e e_i a diferença entre o valor observado e o valor previsto pelo modelo, dado por $e_i = y_i - f(\mathbf{x}_i)$. Pode-se calcular a função de perda para os dados de entrada, sendo interpretada como a perda média da regressão, calculada como:

$$\mathcal{L}(\mathbf{y}, \mathbf{f}|q) = \frac{1}{N}\sum_{j=1}^{N} \mathcal{L}(y_i - f(\mathbf{X_i})|q).$$

A exponenciação ao negativo dessa função de perda média gera a distribuição de Laplace assimétrica. Para essa distribuição, a área sobre o gráfico à esquerda de zero resulta no quantil q requerido. Um resultado interessante advindo da função é ao se utilizar o valor de q igual a 0,5, a função de perda estima a mediana e seu valor é equivalente ao Erro Médio Absoluto (MAE), Abeywardana (2018).

Junto a regressão quantílica, foi-se utilizado a técnica de interpolação por Splines Monotônico. Sua utilidade é de transformar a função de distribuição acumulada, obtida como resultado do modelo quantílico, $\hat{F}(x_k), k = 1, \ldots, n$, em uma função contínua no espaço desejado. No caso, suavizar a curva da distribuição acumulada gerada pelos modelo quantílico, deixando de ser discretizado.

O método de interpolação Splines Monotônico é apresentado por Fritsch e Carlson (1978), sendo uma interpolação unidimensional que utiliza de interpolantes cúbicos por partes, uma classe especial de Splines. Com isso, interpola-se o valor y', tendo (x_i, y_i), $i = 1, \cdots, n$, como os pares a serem a interpolados, da seguinte forma:

$$y'_i = 3(h_{i-1} + h_i)\left(\frac{2h_i + h_{i-1}}{d_{i-1}} + \frac{h_i + 2h_{i-1}}{d_i}\right)^{-1} \mathbb{1}\left(sign(d_{i-1}) = sign(d_i)\right),$$

na qual $h_i = x_{i+1} - x_i$ e $d_i = \frac{y_{i+1}-y_i}{h_i}$, e $\mathbb{1}\left(sign(d_{i-1}) = sign(d_i)\right)$ sendo uma função indicadora que retorna 1 para quando o sinal da distância i é igual ao sinal da distância anterior, *i-1*.

Segundo Morellato (2014), obtém-se dessa fórmula a média harmônica ponderada das inclinações. O método monotônico garante que a inclinação em cada ponto seja sempre positiva, tendo a função resultante o caráter crescente.

A regressão quantílica utilizada neste estudo foi ajustada via redes neurais profundas, com o objetivo de tornar o ajuste da regressão mais robusto, flexível e adequado para uma classe mais ampla de modelos.

Para encontrar o melhor modelo, computa-se a partir da menor função de perda. Apesar desse modelo já ser considerado um modelo não linear, o uso das redes neurais acrescenta ainda mais a esse efeito.

O uso das redes neurais tem grande relevância por conseguirem descrever as relações entre a variável resposta e as variáveis explicativas, seja de forma linear ou não, naquela que melhor se ajusta aos dados reais.

Uma rede neural é composta por camadas de entrada (preditores), camadas ocultas (aplicação dos pesos e funções de ativação não linear) e camada de saída (previsão). As quantidades de camadas, funções de ativação e número de neurônios dependem de caso para caso, mas quanto maior a quantidade delas mais oneroso computacionalmente o algoritmo será. Essa determinação pode ser feita via validação cruzada e erro quadrático médio.

O uso das redes neurais dentro da regressão quantílica faz com que os quantis sejam ajustados simultanenamente. Isso significa que a camada de saída terá um neurônio para cada quantil que será estimado. Logo, obtém-se de apenas uma regressão quantílica os quantis que modelam a distribuição. Foram utilizados, para a modelagem dos métodos, 121 quantis, dispersos igualmente no espaço 0 a 1, resultando, assim, em 121 camadas de saída da rede neural. Uma vantagem de se utilizar essa modelagem é que o tempo

computacional não cresce linearmente.

2.5 Amostrador de Gibbs Aproximado sem a função de Verossimilhança proposto por Rodrigues, Nott e Sisson (2019)

Rodrigues, Nott e Sisson (2019) propuseram um algoritmo para gerar amostras aproximadas de distribuições condicionais completas, que são intratáveis, por meio de um Amostrador de Gibbs aproximado. Neste algoritmo a estrutura de covariância da distribuição a posteriori é capturada naturalmente pelo método.

Há um ganho considerável neste método em tornar o modelo mais simples e fácil de estimar seus parâmetros, principalmente para distribuições de alta dimensão.

O método proposto por Rodrigues, Nott e Sisson (2019), chamado de Amostrador de Gibbs Aproximado sem a função de Verossimilhança, segue a mesma proposta do algoritmo simples do ABC, apresentado na seção 2.1, utilizando modelos de regressão para modelar as distribuições condicionais completas e o aprimora incluindo o Amostrador de Gibbs. Logo, o conhecimento da distribuição exata dos dados é desnecessária, podendo se utilizar de amostras aproximadas.

Ao implementar o Amostrador de Gibbs Aproximado, cada parâmetro é gerado a partir da aproximação à sua respectiva distribuição condicional completa obtida pelo modelo de regressão $\theta_d^{(m)}|(\boldsymbol{s}_{obs}, \boldsymbol{\theta}_{-d}) \sim f(\theta_d|\hat{\boldsymbol{\beta}}_d^+, g_d(\boldsymbol{s}_{obs}, \boldsymbol{\theta}_{-d}))$, para $d = 1, \ldots, D$.

O Amostrador de Gibbs gera a distribuição condicional a posteriori, $\pi(\boldsymbol{\theta}|\boldsymbol{s}_{obs})$, se $f(\theta_d|\hat{\boldsymbol{\beta}}_d^+, \boldsymbol{s}_{obs}, \boldsymbol{\theta}_{-d}) = \pi(\theta_d|\boldsymbol{s}_{obs}, \boldsymbol{\theta}_{-d})$ para todo d.

Segundo Rodrigues, Nott e Sisson (2019), o método apresenta limitações quanto a mistura lenta da cadeia de Markov, especialmente em modelos com fortes correlações na distribuição a posteriori, fato que é explicado pelo uso do Amostrador de Gibbs que herda essa condição.

A abordagem também é sensível a problemas em que as distribuições condicionais

completas aproximadas não são compatíveis com a existência de uma distribuição conjunta única.

Outro problema se dá pelo uso de modelos ajustados para a média e variância para aproximar as distribuições condicionais completas. Com isso, mesmo que sejam adotados modelos mais flexíveis, como o uso de redes neurais, há uma suposição implícita de que a forma da distribuição é fixa em uma vizinhança de s_{obs}. O que pode limitar a cobertura do método para quando se trabalhar com dados mais complexos.

Uma forma de melhorar esse algoritmo é ajustar modelos localmente adequados em cada iteração do Gibbs, porém é necessário ajustar milhares de modelos de regressão, o que pode se tornar inviável computacionalmente.

O método de Rodrigues, Nott e Sisson (2019) é implementado no Algoritmo 2 ao final desta seção.

2.6 Método de Santos (2021)

Para melhorar o método de Rodrigues, Nott e Sisson (2019), Santos (2021) propôs duas inovações, a primeira utilizando a descorrelação prévia dos parâmetros para acelerar a convergência e aumentar a velocidade de mistura, fazendo com que a busca feita pelo algoritmo consiga explorar completamente o suporte da distribuição alvo mais rapidamente. Com isso, é possível trabalhar, também, no espaço transformado dos dados, para caso haja forte autocorrelação entre os parâmetros.

A descorrelação dos parâmetros $\boldsymbol{\theta}_d$ é feita pela técnica de PCA ponderada, tendo os pesos calculados por uma rodada preliminar do ABC. Com isso, gera-se um novo vetor de parâmetros transformados $\boldsymbol{\theta}_d$ que serão utilizados na estimação dos modelos de regressão.

A segunda inovação de Santos (2021) é modelar toda a distribuição alvo usando estimadores flexíveis de densidade condicional, como aqueles baseados na estimação conjunta dos quantis via redes neurais profundas com correção Splines Monotônico, diferentemente

do encontrado em Rodrigues, Nott and Sisson (2019) que utiliza modelos que estimam a média e variância. Com isso, assume-se que a forma das distribuições condicionais completas não é necessariamente fixa em uma vizinhança de s_{obs}.

O uso de interpolação monotônica com correção Splines também será aplicado para gerar uma função contínua no espaço para os quantis obtidos pelo modelo de regressão quantílica via redes neurais.

Essa abordagem permite modelar os quantis explicitamente fornecendo uma completa descrição da distribuição e se destaca no uso de dados não-lineares e complexos.

Vale mencionar que este método ainda se utiliza do Amostrador de Gibbs para geração de amostras da distribuição a posterior.

O tempo computacional em comparação aos métodos foi o mesmo, segundo Santos (2021), mas a velocidade de mistura da cadeia de Markov foi consideravelmente maior, muito por conta de eliminação da autocorrelação imposta pela técnica de PCA ponderada.

O método de Santos (2021) é implementado no Algoritmo 3 ao final desta seção.

Algoritmo 2: Amostrador de Gibbs aproximado, sem a função de verossimilhança - Método Original

Entrada:

Um conjunto de dados observados ($\boldsymbol{X}_{obs}$);
Uma distribuição a priori $\pi(\boldsymbol{\theta})$ e um modelo generativo intratável $p(\boldsymbol{X}|\boldsymbol{\theta})$;
Uma distribuição $b(\boldsymbol{\theta})$ descrevendo uma região de alta densidade da distribuição
posteriori;
Um vetor observado de medidas resumo $s_{\boldsymbol{X}_{obs}=S(\boldsymbol{X}_{obs})}$;
Um kernel de suavização $K_h(u)$ com parâmetro de escala $h > 0$;
Um inteiro positivo N definindo o número de amostras ABC;
Um inteiro positivo M definindo o número de iterações do amostrador Gibbs;
Uma coleção de modelos de regressão $f(\theta_d|\beta_d^+, g_d(S, \theta_{-d}))$ para aproximar cada distribuição condicional completa $\pi(\theta_d|s_{\boldsymbol{X}_{obs},\theta_{-d})}$ para $d = 1, \ldots, D$.

// Simulação de Dados Sintéticos

para $i = 1, 2, \cdots, N$ **faça**

1.1 Gere $\theta^{(i)} \sim b(\boldsymbol{\theta})$;
1.2 Gere $X^{(i)} \sim p(\boldsymbol{X}|\theta^{(i)})$ do modelo;
1.3 Calcule as medidas resumo $s^{(i)} = S(\boldsymbol{X}^{(i)})$;
1 4 Calcule os pesos da amostra $w^{(i)} \propto K_h(\|s^{(i)} - s_{obs}\|)\pi(\theta)/b(\theta)$;

fim

Inicializar $\tilde{\theta}^{(0)} = (\tilde{\theta}_1^{(0)}, \ldots, \tilde{\theta}_D^{(0)})^\top$;

// Estimação dos Modelos

para $d = 1, 2, \cdots, D$ **faça**

2.1 Ajuste um modelo de regressão adequado $\theta_d|(S,_{-d}) \sim f(\theta_d|\beta_d^+, g_d(S, \theta_{-d}))$, de modo a $f(\theta_d|\hat{\beta}_d^+, g_d(s_{obs}, \theta_{-d}))$ aproximar localmente a distribuição condicional completa $p(\theta_d|s_{obs}, \theta_{-d})$;

fim

// Aproximação de Gibbs

para $m = 1, 2, \cdots, M$ **faça**

para $d = 1, 2, \cdots, D$ **faça**

3.1 $\theta^\star_{-d} = (\tilde{\theta}_1^{(m)}, \ldots, \tilde{\theta}_{d-1}^{(m)}, \tilde{\theta}_{d+1}^{(m-1)}, \ldots, \tilde{\theta}_D^{(m-1)})^\top$ o vetor que contém os valores atualizados de $\tilde{\theta}_j^{(\cdot)}$, $j \neq d$;
3.2 Atualização do Gibbs: Amostre $\tilde{\theta}_d^{(m)}|(s_{obs}, \theta^\star_{-d}) \sim f(\theta_d|\hat{\beta}_d^+, g_d(s_{obs}, \theta^\star_{-d}))$;

fim

fim

Saída:

O vetor contendo amostras aproximadas de Gibbs para os parâmetros.

Algoritmo 3: Amostrador de Gibbs aproximado, com inovações - Santos (2021)

Entrada:

Um conjunto de dados observados (x_{obs});

Um vetor observado de medidas resumo $s_{obs} = \boldsymbol{S}(x_{obs})$;

[Opcional] Um kernel de suavização $K_h(u)$ com parâmetro de escala $h > 0$;

Uma função $\psi : [0,1] \longrightarrow$R,x$\longmapsto\psi$(x);

Um modelo generativo $\pi(\theta)p(X|\theta)$;

Um inteiro positivo N definindo o número de amostras sintéticas geradas;

Um inteiro positivo M definindo o número de iterações do amostrador Gibbs;

Um vetor $q = (q_1, q_2, \ldots, q_k)$, em que, $0 < q_k < 1$, que define os quantis a serem estimados;

Uma distribuição adequada de $b(\theta)$

// Simulação de Dados Sintéticos

para $i = 1, 2, \cdots, N$ **faça**

1.1 Gere $\theta^{(i)} \sim b(\theta)$;

1.2 Gere $X^{(i)} \sim p(X|\theta^{(i)})$ do modelo;

1.3 Calcular as medidas resumo $s^{(i)} = \boldsymbol{S}(X^{(i)})$;

fim

// [Opcional] Descorrelação via PCA Ponderada

2.1 Calcular os parâmetros na escala transformada $\check{\theta}$ para todo N, e substituir θ por $\check{\theta}$ em (3.1 e 4.2.1);

// Estimação dos Modelos Quantílicos

3.1 Ajustar modelos de regressão quantílica via redes neurais $\boldsymbol{\tau_q}(\theta_d)|g_d(S, \theta_{-d})$ para aproximar cada distribuição condicional completa $\pi(\theta_d|s_{obs}, \theta_{-d})$ para $d = 1, \ldots, D$;

// Aproximação Gibbs sampling

4.1 Inicializar $\tilde{\theta}^{(0)} = (\tilde{\theta}_1^{(0)}, \ldots, \tilde{\theta}_D^{(0)})^\top$;

para $m = 1, 2, \cdots, M$ **faça**

para $d = 1, 2, \cdots, D$ **faça**

4.1.1 Obtenha $\hat{\tau}_q(\theta_d)|\, g_d(_{obs}, \theta^\star_{-d})$, em que $\theta^\star_{-d} = (\tilde{\theta}_1^{(m)}, \ldots, \tilde{\theta}_{d-1}^{(m)}, \tilde{\theta}_{d+1}^{(m-1)}, \ldots, \tilde{\theta}_D^{(m-1)})^\top$ o vetor que contém os valores atualizados de $\tilde{\theta}_j^{(\cdot)}$, $j \neq d$;

4.1.2 Ajustar um Splines Monotônico da forma $\hat{F}(\cdot)$ sobre os pontos $(\psi(q), \hat{\tau}_q)$;

4.1.3 Gerar um valor aleatório $u \sim U[0,1]$;

4.1.4 Aplique o método da transformada inversa $\tilde{\theta}_d^{(m)} = \hat{F}^{-1}(\psi(u))$;

fim

fim

// [Opcional] Mudança de Escala

4.2 Realizar transformação inversa de $\check{\theta}$, para retornar à escala original θ;

Saída:

A cadeia $\{\boldsymbol{\theta}_t | t = 0, 1, \cdots, M\}$ contendo os valores gerados.

Capítulo 3

Método proposto

O método proposto neste trabalho junta a ideia de fatoração da distribuição a posteriori, proposto por Rodrigues, Nott e Sisson (2019), com a ideia de Santos (2021) de usar Regressão Quantílica via redes neurais e interpolação Splines Monotônica.

Sua inovação consiste em aproximar a distribuição a posteriori via decomposição por fatorações, ao invés de aproximar as distribuições condicionais completas e incorporá-las em um Amostrador de Gibbs aproximado, como feitas nos trabalhos de Rodrigues, Nott e Sisson (2019) e de Santos (2021).

As densidades condicionais ainda são aproximadas por modelos de regressão quantílica com redes neurais, mas as amostras são geradas independentemente, sem executar uma Cadeia de Markov. Portanto, não há necessidade geral de descorrelacionar os parâmetros, embora qualquer transformação possa simplificar (ou complicar) a forma das densidades condicionais a serem estimadas.

A decomposição da distribuição a posteriori pode ser feita de várias formas, como na forma de blocos, como:

$$p(\theta_1, \ldots, \theta_n | \boldsymbol{X}) = p(\theta_1, \theta_2, \theta_3, \ldots, \theta_m | \boldsymbol{X}) \times p(\theta_{m+1}, \ldots, \theta_n | \theta_1, \theta_2, \theta_3, \ldots, \theta_m, \boldsymbol{X}),$$

no qual $m < n$.

Outra forma de decomposição e que será utilizada no algoritmo proposto neste trabalho, é a seguinte:

$$p(\theta_1, \ldots, \theta_n|\boldsymbol{X}) = p(\theta_1|\boldsymbol{X})p(\theta_2|\theta_1, \boldsymbol{X}) \ldots p(\theta_n|\theta_1, \ldots, \theta_{n-1}, \boldsymbol{X}).$$

O efeito da escolha dessa ordenação na fatoração será investigado a fim de descobrir se existem formas adequadas de definí-la.

O método segue os passos de Rodrigues, Nott e Sisson (2019), assim como de Santos (2021), com a diferença de realizar aproximações da distribuição a posteriori via fatorações ao invés do Amostrador de Gibbs. O algoritmo 4 elucida seu funcionamento.

Algoritmo 4: Aproximação da distribuição a posteriori via fatorações - método proposto

Entrada:

Um conjunto de dados observados ($\boldsymbol{X_{obs}}$);

Um vetor observado de medidas resumo $s_{obs} = \boldsymbol{S}(x_{obs})$;

Uma função $\psi : [0,1] \longrightarrow$R,x$\longmapsto \psi(x)$;

Um modelo generativo $\pi(\theta)p(\boldsymbol{X}|\boldsymbol{\theta})$;

Um inteiro positivo M definindo o número de amostras sintéticas a serem geradas;

Um vetor $q = (q_1, q_2, \ldots, q_k)$, em que, $0 < q_k < 1$, que define os quantis a serem estimados;

Escolha da forma da fatoração, considerando
p$(\theta_1, \ldots, \theta_n|X) = p(\theta_1|X)p(\theta_2|\theta_1, X) \ldots p(\theta_n|\theta_1, \ldots, \theta_{n-1}, X)$

// Simulação de Dados Sintéticos

para $i = 1, 2, \cdots, N$ **faça**

1.1 Gere $\theta^{(i)} \sim b(\theta)$ de alguma distribuição adequada $b(\theta)$;

1.2 Gere $X^{(i)} \sim p(X|\theta^{(i)})$ do modelo;

1.3 Calcular as medidas resumo $s^{(i)} = \boldsymbol{S}(X^{(i)})$;

fim

// Estimação dos Modelos Quantílicos

3.1 Ajustar modelos de regressão quantílica via redes neurais $\boldsymbol{\tau_q}(\theta_d)|g_d(S,)$ para aproximar cada distribuição condicional contida na fatoração da distribuição a posteriori; *// Aproximação via fatoração*

4.1 Obtenha $\hat{\tau}_q(\theta_1)|g_d(s_{obs}, \theta^{\star}_{-d})$, em que $\theta^{\star}_{-d} = (\tilde{\theta}_1^{(m)}, \ldots, \tilde{\theta}_{d-1}^{(m)}, \tilde{\theta}_{d+1}^{(m-1)}, \ldots, \tilde{\theta}_D^{(m-1)})^{\top}$ o vetor que contém os valores atualizados de $\tilde{\theta}_j^{(\cdot)}$, $j \neq d$;

4.1.2 Ajustar um Splines Monotônico da forma $\hat{F}(\cdot)$ sobre os pontos $(\psi(q), \hat{\tau}_q)$;

4.1.3 Gerar um valor aleatório $u \sim U[0,1]$;

para $m = 1, \cdots, M$ **faça**

4.1.4 Aplique o método da transformada inversa para obter $\tilde{\theta}_1^m = \hat{F}^{-1}(\psi(u))$;

fim

para $m = 1, \cdots, M$ **faça**

para $d = 2, \cdots, D$ **faça**

4.2.1 Obtenha $\hat{\tau}_q(\theta_d)|g_d(S_{obs}, \theta^{\star}_d)$, em que $\theta^{\star}_d = \theta_1^m, \ldots, \theta_{d-1}^m$;

4.2.2 Ajustar um Splines Monotônico da forma $\hat{F}(\cdot)$ sobre os pontos $(\psi(q), \hat{\tau}_q)$;

4.2.3 Gerar um valor aleatório $u \sim U[0,1]$;

4.2.4 Aplique o método da transformada inversa $\theta_d^m = \hat{F}^{-1}(\psi(u))$;

fim

fim

Saída:

Um conjunto de amostras $\{\boldsymbol{\theta}_t | t = 1, \cdots, N\}$ contendo os valores gerados.

Capítulo 4

Estudos simulados

4.1 Normal Bivariada

O presente exemplo consiste de três implementações de algoritmos para o problema de obtenção do vetor de médias da distribuição Normal Bivariada com matriz de covariâncias conhecida. O objetivo desta seção é realizar uma aplicação inicial de derivações dos algoritmos discutidos no Capítulo 3 em uma distribuição de baixa complexidade e comparar seus resultados.

Os três algoritmos utilizados fazem uso dos mesmos dados gerados. Portanto, o procedimento inicial dos métodos se dá pela geração e obtenção dos mesmos dados. As estatísticas-resumo são dadas pela média da distribuição de cada variável em $\boldsymbol{X}$. Simulados os dados, aplica-se cada algoritmo e analisa-se seus comportamentos. Os algoritmos são:

i Implementação 1 - proposto por Rodrigues, Nott e Sisson (2019), descrito no algoritmo 2 da seção 2, faz uso de modelos de regressão normal para gerar as distribuições condicionais completas $(\theta_1|s_1, s_2, \theta_2)$ e $(\theta_2|s_1, s_2, \theta_1)$. A partir daí, utiliza-se o Amostrador de Gibbs para gerar amostras da distribuição a posteriori de cada θ.

ii Implementação 2 - proposto por Santos (2021), descrito no algoritmo 3 proposto na

seção 2, com a alteração de que não é utilizado o modelo de regressão quantílica via redes neurais, mas sim, modelos de regressão normal. Com isso, toda a parte de suavização via Splines Monotônico também não é utilizada. O procedimento, então, consiste em realizar a transformação PCA para descorrelacionar os parâmetros da distribuição gerada, logo após aplicação do algoritmo ABC. Depois, utiliza-se os modelos de regressão normal para estimar as distribuições condicionais completas, essas entram no Amostrador de Gibbs para gerar a distribuição a posteriori de θ. Por fim, reverte-se os valores transformados pela PCA da distribuição da posteriori obtida.

iii Implementação 3 - proposto na seção 3 e objeto deste estudo, mas com alterações quanto ao modelo de regressão utilizado. Nesse, substitui o modelo de regressão quantílica via redes neurais e suavização dos dados via Splines Monotônico pelo modelo de regressão normal. Após ajuste da regressão linear, aproxima-se os valores da distribuição a posteriori de θ_1 e θ_2 por fatorações da distribuição a posteriori. Esse algoritmo não faz uso da amostra transformada, fazendo com que seus dados de entrada sejam os mesmos utilizados na Implementação 1.

Para geração dos dados, foi-se utilizado uma variável aleatória, $\mathbf{X}$, que segue uma distribuição Normal Bivariada com média $\boldsymbol{\mu} = (\mu_1, \mu_2)$ e matriz de covariância conhecida, $\boldsymbol{\Sigma}$. O vetor de médias é dado por uma distribuição a priori da distribuição Uniforme(-10,10), sendo:

$$\boldsymbol{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{4.1}$$

$$\boldsymbol{\mu} \overset{\text{i.i.d.}}{\sim} U[-10, 10]. \tag{4.2}$$

Para a distribuição Normal Bivariada, pode-se comparar os resultados das distribuições obtidas com as respectivas distribuições exatas, por conta da função de verossimilhança ser facilmente computada.

Usando as propriedades da distribuição Normal Multivariada, a distribuição condicional completa para μ_1 é proporcional a

$$\mu_1|\mu_2 = \mu_2^{'} \sim N\left(\mu_1^* + \frac{\rho}{\sigma_{22}^*}(\mu_2^{'} - \mu_2), \sigma_{11}^* - \frac{\rho}{\sigma_{22}^*}\right) \tag{4.3}$$

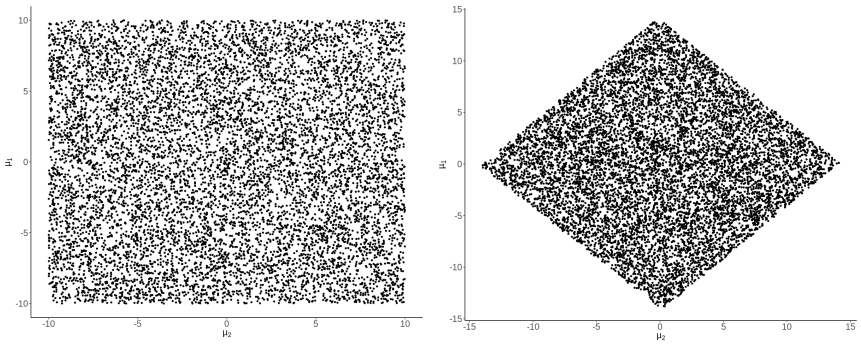
sendo μ_1 e μ_2 os parâmetros de média, ρ a correlação entre X_1 e X_2, e σ_{11} e σ_{22} as variâncias de X_1 e X_2.

O processo descrito acima é análogo para a distribuição condicional completa de μ_2.

Para o estudo de simulação, foram utilizados os parâmetros: $\rho = 9/10$, $\sigma_1 = \sigma_2 = 1$, $s_{obs} = (5/2, 5/2)$ e $\boldsymbol{\mu} \overset{\text{i.i.d.}}{\sim} U[-10, 10]$.

Com isso, foram geradas N = 10.000 amostras considerando $b(\mu) = \pi(\mu)$ para obter $(\mu_1^i, \mu_2^i, s_1^i, s_2^i)$, i = 1, ..., N. Depois, procedeu-se com a realização do ABC pelo método *Loc-linear* com taxa de aceitação de 0,1, sendo este método do ABC o descrito na seção 2.1.1, onde ajusta-se uma regressão linear local para aproximar $s(y)$ de $s(y_{obs})$.

Na Figura 4.1 são mostrados os gráficos da distribuição das amostras da distribuição a posteriori geradas e depois de passarem pela transformação PCA ponderada, gráfico (b). Ambos os gráficos representam os dados de entrada nas implementações utilizados nesta seção, sendo (a) utilizado para as Implementação 1 e 3 e (b) para a Implementação 3.



(a) Amostra da distribuição a priori (b) Amostra da distribuição a priori transformada

Figura 4.1: Amostras da distribuição a priori segundo cada método

Pela Figura 4.1 (a) é possível observar as amostras da distribuição a priori geradas

através de uma distribuição Uniforme[-10,10] para cada μ_i, dado pelo espalhamento aleatório dos dados sobre a área do gráfico que compreende o intervalo -10 e 10.

Esse intervalo foi definido a partir de um ABC piloto que foi rodado para identificar a região de alta densidade da distribuição a posteriori. Usar intervalos muito grandes que estão fora do escopo factível dos dados reais fará com que o ABC rejeite a maioria dos conjuntos das amostras geradas, sendo um esforço computacional gasto desnecessariamente.

Já (b), indica esses mesmos dados quando aplicados em uma transformação, PCA ponderada. É possível observar uma compressão na distribuição dos dados, em formato de losango. A transformação realizada tem por objetivo descorrelacionar os parâmetros gerados da distribuição a posteriori.

Com os dados de entrada gerados, parte-se para a aplicação dos algoritmos.

A primeira e segunda implementações fazem uso do Amostrador de Gibbs, assim, utilizou-se 5000 iterações cada um para produzir os resultados da distribuição a posteriori, utilizando como valor inicial $(\mu_1, \mu_2) = (0, 0)$.

Para a terceira, que faz uso das fatorações, gerou-se 5000 amostras independentes da distribuição a posteriori.

Os gráficos abaixo mostram o comportamento das distribuições a posteriori θ_1 e θ_2 encontrados para cada um dos algoritmos. Para os gráficos (b), (c) e (d) da Figura 4.2, são apresentadas as distribuições a posteriori conjuntas encontradas em cada implementação e, em (a), a distribuição a posteriori encontrada pelo algoritmo ABC do pacote *abc* do R. Os contornos em vermelho nesses gráficos representam as curvas de nível e os pontos a distribuição conjunta a posteriori.

O objetivo dos gráficos (a), (b), (c) e (d) é comparar a razoabilidade dos métodos, a partir das distribuições a posteriori obtidas e a aproximação com a verdadeira densidade.

A obtenção da distribuição a posteriori via método ABC, apresentado no gráfico (d) serve como uma forma de comparação dos valores finais das distribuições, podendo ser

interpretado como o modelo ideal da distribuição a posteriori para esse exemplo. Além disso, a Figura 4.2 apresenta as densidades marginais das distribuições a posteriori de θ_1 e θ_2 para cada algoritmo, em (e) e (f), juntamente com a distribuição real exata do exemplo.

Os comportamentos das distribuições a posteriori conjuntas tiveram um formato de elipse, principalmente se observada as curvas de nível, em torno do centro, sendo esse o ponto (2,5; 2,5). Sendo esse o ponto cuja as estatísticas-resumo são as corretas para distribuição real. Logo, todos os algoritmos funcionam bem pois têm mesmo comportamento do algoritmo ABC.

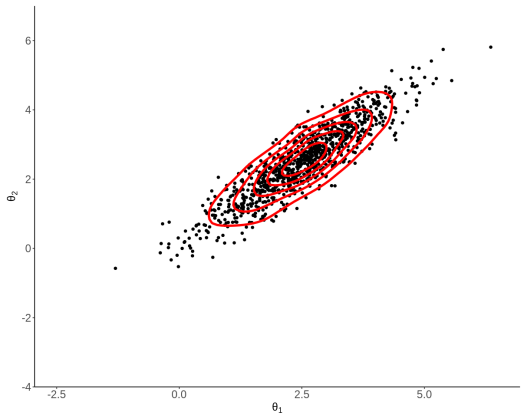
Vale destacar que o gráfico (a) da Figura 4.2 tem menor massa de pontos pois tem menos quantidade de dados gerados. Isso se deve a rejeição de dados realizada pelo algoritmo ABC, tendo utilizado uma tolerância de rejeição de 10%.

Portanto, como visto nos gráficos acima todos os algoritmos funcionam pois se parecem entre si e parecem com as amostras aceitas pelo algoritmo de rejeição ABC. Essa afirmação vale tanto na comparação das densidades conjuntas quanto nas densidades marginas, na qual são comparados os valores com a distribuição real exata.

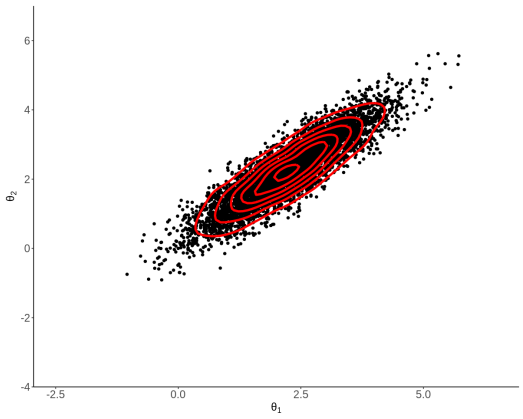
Para avaliar as características de cada algoritmo, são apresentados na Figura 4.3 a mistura da cadeia, denotado pelo caminho percorrido pelas primeiras 50 iterações realizadas pelo Amostrador de Gibbs (implementações 1 e 2) e fatorações da distribuição a posteriori (implementação 3). Além disso, também são apresentadas as funções de autocorrelação para cada algoritmo.

A mistura da cadeia ilustra os saltos que o algoritmo proporciona para movimentação de um ponto para seu sucessivo. Isso é importante para mostrar qual algoritmo tem maior mistura nas observações (maiores saltos) do que outros. Isso pode influenciar no caminho e tempo computacional necessário para convergência ao valores corretos da distribuição.

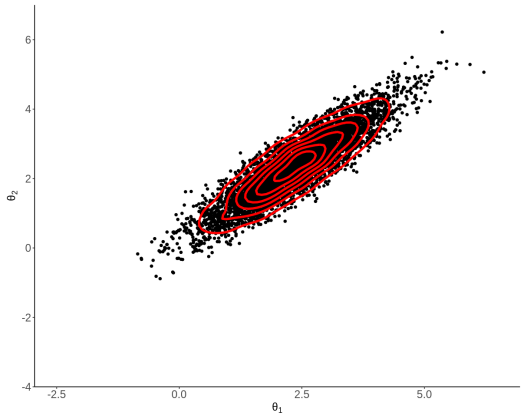
Já as funções de autocorrelação indicam o quão correlacionadas duas observações sucessivas estão. Valores altos apontam que os parâmetros estão autocorrelacionados, o que



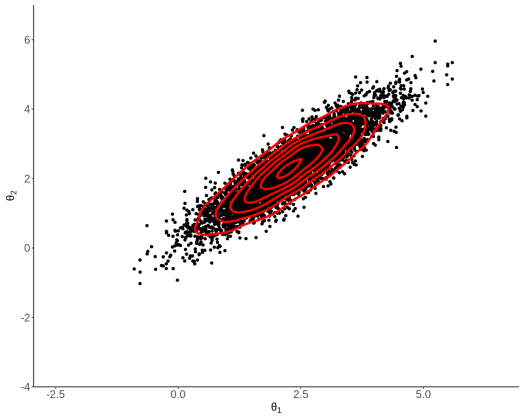
(a) Amostras aceitas pelo algoritmo de rejeição ABC



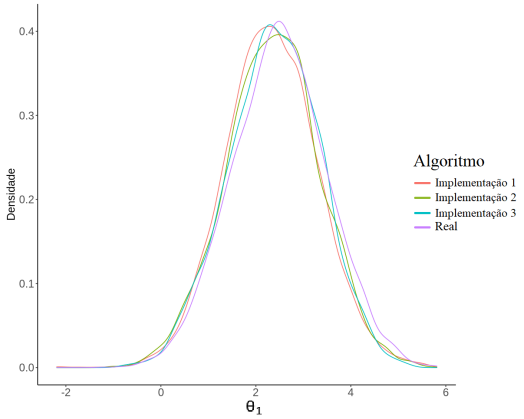
(b) Distribuição a posteriori da Implementação 1



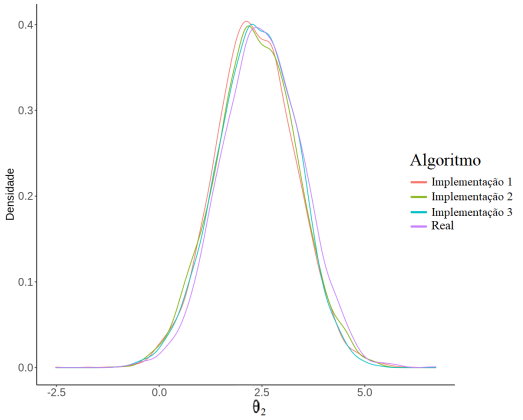
(c) Distribuição a posteriori da Implementação 2



(d) Distribuição a posteriori da Implementação 3



(e) Densidade marginal da distribuição a posteriori de θ_1 para cada implementação



(f) Densidade marginal da distribuição a posteriori de θ_2 para cada implementação

Figura 4.2: Distribuições conjuntas a posteriori e densidades marginais.

pode impactar na mistura.

Os gráficos, da Figura 4.3(a), (c) e (e) mostram as misturas das cadeias para as 50 primeiras iterações do Amostrador de Gibbs e os gráficos (b), (d) e (f) mostram as funções de autocorrelação (ACF) de θ_1 para cada uma das implementações.

Para economia de espaço é apresentado somente o comportamento do ACF para θ_1, pois o comportamento para θ_2 foi equivalente.

Pelos gráficos de mistura da Figura 4.3, é possível observar um comportamento de maior variabilidade, dada pelas distâncias entre pontos consecutivos da cadeia, para as misturas nas Implementação 2 e 3 e bem menor na Implementação 1.

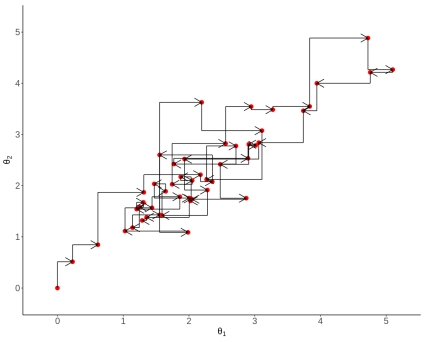
Vale destacar que foram testadas as iterações em diferentes partes da cadeia (começo, meio e final da cadeia). O comportamento nessas partes, para as três implementações, foi parecido com os encontrados para as 50 primeiras iterações, como mostrado nos gráficos (a), (c) e (e). Isso se deve ao fato de que as distribuições condicionais, para cada algoritmo, são as mesmas ao longo de todo o processo.

Percebe-se, ainda, o efeito da descorrelação na função de autocorrelação (ACF), que, para a Implementação 1 (gráfico (b) da Figura 4.3) apresenta a maioria dos lags acima do limite, mostrando que há autocorrelação na observações consecutivas geradas por esse método.

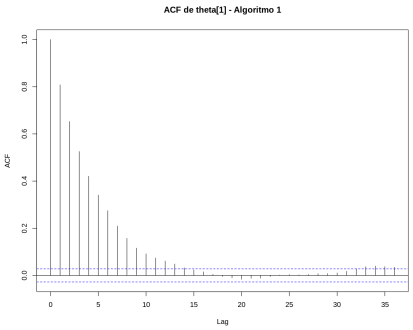
Já para as funções de ACF das implementações 2 e 3, vistas nos gráficos (d) e (f), é possível observar que os lags se encontram dentro dos limites aceitos, linha tracejada. Isso indica ausência de autocorrelação das observações consecutivas geradas por cada algoritmo.

Logo, por não serem autocorrelacionadas, pode-se inferir que foram geradas efetivamente amostras independentes da distribuição a posteriori. O que não ocorre na Implementação 1, em que há uma forte correlação nos lags iniciais, com um decaimento exponencial (característico de um modelo auto-regressivo).

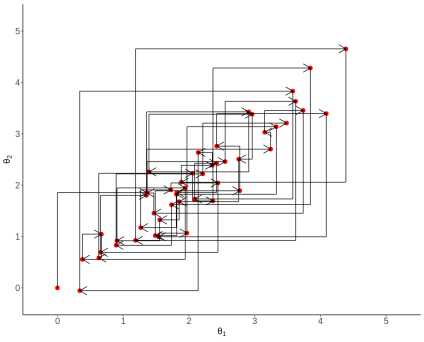
Pode-se discorrer, então, que o Amostrador de Gibbs sofre com a autocorrelação.



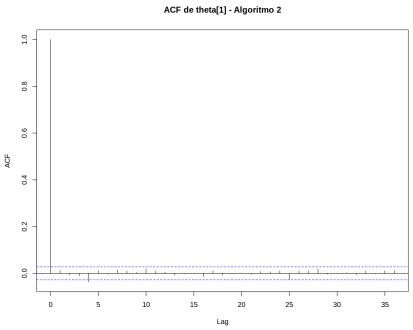
(a) Mistura da cadeia das 50 primeiras iterações do Amostrador de Gibbs para a Implementação 1



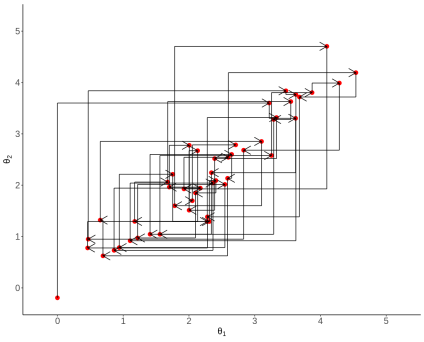
(b) Função de autocorrelação do Implementação 1



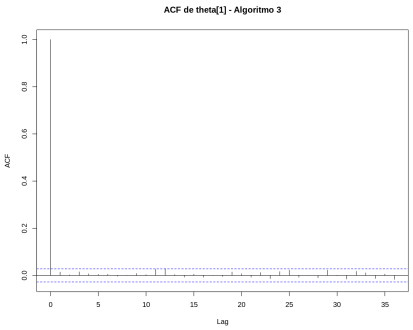
(c) Mistura da cadeia das 50 primeiras iterações do Amostrador de Gibbs para a Implementação 2



(d) Função de autocorrelação da Implementação 2



(e) Mistura da cadeia das 50 primeiras iterações do Amostrador de Gibbs para a Implementação 3



(f) Função de autocorrelação da Implementação 3

Figura 4.3: Mistura da cadeia e função de autocorrelação

Com isso, utiliza-se o Gibbs no espaço transformado, como uma tentativa de minimizar a autocorrelação, mas que não é garantida por esse método (Implementação 2).

Quando se usa a transformação PCA ponderada, as amostras são linearmente não correlacionadas. Portanto, a Implementação 2 conseguiu reduzir a autocorrelação, mas não necessariamente ajudará na velocidade da mistura, sobretudo quando houver correlações não lineares. Além disso, os θ's continuam sendo dependentes do seu estado passado. Já a Implementação 3, pela sua composição por fatorações da distribuição a posteriori, pode até ser amostrado em paralelo, pois uma não depende da outra.

Para avaliar a velocidade de convergência dos algoritmos a partir do tamanho de amostras geradas pelo Amostrador de Gibbs (Implementação 1 e 2) ou das fatorações da distribuição a posteriori (Implementação 3), foram calculados os erros quadráticos médios (MSE) encontrados em cada algoritmo para diferentes tamanhos de amostra gerados. O MSE se dá pela média das diferenças quadráticas entre o vetor de estatísticas-resumo aproximado e o real ponto da distribuição (2,5;2,5).

Os resultados são apresentados no gráfico da Figura 4.4.

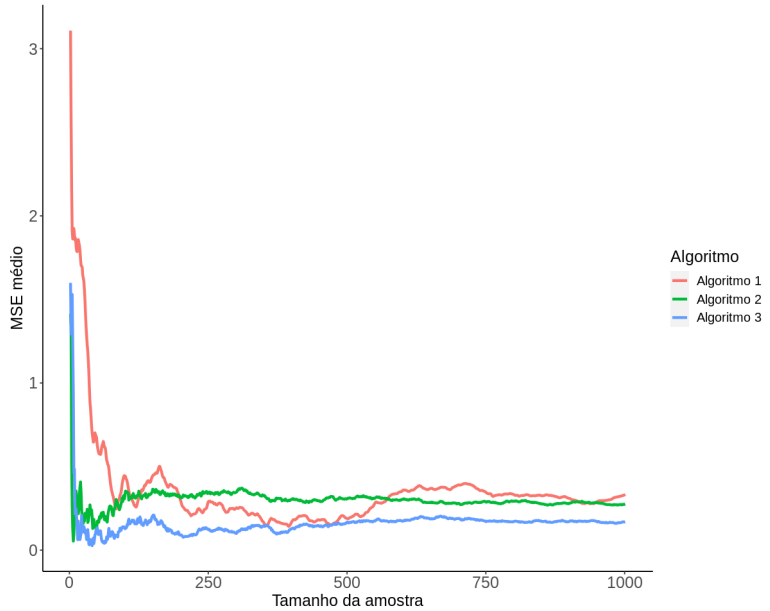


Figura 4.4: Velocidade de convergência das amostras de tamanho até 1.000

Pela Figura 4.4, é possível observar que a Implementação 3, aproximação por fatorações da posterior, apresentou menor MSE encontrado para todos os tamanhos de amostra até 1.000, se comparados com seus pares.

Há uma estabilização dos valores de MSE a partir do tamanho 125, aproximadamente, para as implementações 2 e 3, e a partir do tamanho 750 para a Implementação 1. Indicando assim, que a convergência para a real distribuição é mais rápida e necessita de menos iteração nos algoritmos 2 e 3.

4.2 *Twisted Gaussian*

Considere, agora, um modelo determinístico da distribuição Twisted Normal dado pela variável $Y = \theta_1 + \theta_2^2$, com $\boldsymbol{\theta} = (\theta_1, \theta_2)^T$ e considerando que θ_1 e θ_2 são parâmetros da distribuição a priori com cada uma tendo distribuição N(0,1).

Para um único ponto observado $y_{obs} = 1$, o resultado da densidade da distribuição a posteriori está concentrado em um conjunto de pontos que satisfaça $\theta_1 = 1 - \theta_2^2$.

As estatísticas-resumo são dadas pela média da distribuição de cada variável em $\boldsymbol{X}$.

Foram três os algoritmos implementados nesta simulação, sendo dois deles advindos do mesmo processo de modelagem (fatorações da distribuição a posteriori), com a alteração na ordem como é fatorada a distribuição a posteriori.

i Implementação 4 - proposto na seção 3, utiliza o algoritmo como mencionado naquela seção. A fatoração da distribuição a posterior tem a ordem de $p(\theta_1) = p(\theta_1|y_{obs})p(y_{obs})$ e $p(\theta_1, \theta_2|y_{obs}) = p(\theta_2|\theta_1, y_{obs})p(\theta_1|y_{obs})$.

ii Implementação 5 - proposto na seção 3 e objeto deste estudo, utiliza inteiramente o algoritmo como mencionado naquela seção. A fatoração da distribuição a posterior tem a ordem invertida da Implementação 4, sendo $p(\theta_2) = p(\theta_2|y_{obs})p(y_{obs})$ e $p(\theta_1, \theta_2|y_{obs}) = p(\theta_1|\theta_2, y_{obs})p(\theta_2|y_{obs})p(y_{obs})$.

iii Implementação 6 - proposto por Santos (2021), descrito no algoritmo 2 da seção 2, que utiliza do Amostrador de Gibbs para gerar amostras da distribuição a posteriori via as distribuições condicionais $\theta_1 = (\theta_1|\theta_2, y_{obs})$ e $\theta_2 = (\theta_2|\theta_1, y_{obs})$, sem o uso da descorrelação prévia dos parâmetros.

A forma de obtenção das densidades condicionais dos três algoritmos passa por um ajuste de regressões quantílicas via redes neurais com transformação Splines Monotônico. O próximo passo a ser realizado é o processo de geração de amostras a distribuição a posteriori que difere para as implementações 4 a 6.

A regressão quantílica via redes neurais utilizada nos algoritmos, descrita na seção 4.3, utiliza em sua arquitetura 7 camadas de 512, 256, 128, 64, 32, 16 e 121 neurônios, respectivamente, em cada camada, totalizando, assim, 1.129 neurônios na rede. O quantitativo de 121 neurônios na última camada representa o total de quantis que foram estimados ao final do modelo, sendo cada um representando um quantil de tamanho 0,82%.

Também há a utilização de *Dropout* na primeira camada, com taxa de 0,4 e *L2-regularization* na segunda camada, com taxa de 0,01, para prevenir o *overfitting*. O otimizador foi o *Adam*, a taxa de aprendizado utilizada foi de 0,2, o tamanho do *Batch Size* de 64 com 100 épocas para realização na base de treinamento.

Para a escolha do tamanho de amostras utilizadas para a distribuição a priori, foram testados tamanhos iguais a 5.000, 100.000 e 1.000.000, sendo eles divididos em 80% para dados de treinamento e 20% para dados de teste. Também, computou-se os tempos computacionais de geração de cada algoritmo para cada tamanho de amostra. O algoritmo ABC também foi utilizado para conferência da convergência das amostras a distribuição a posteriori geradas, em que foi utilizada uma taxa de aceitação de 0,005 e método *Loc-linear*.

Comparou-se os custos a partir de diferentes máquinas, sem GPU e com GPU, ambas advindas da versão Google Colab Pro. A tabela abaixo descreve os tempos computacionais

encontrados. São mostrados nessa tabela os tempos para obtenção de θ_1 e, na mesma célula, de θ_2, separados pelo sinal matemático de mais. Foram mensurados os tempos para cada tipo de máquina, tamanho de amostra e algoritmo.

	Sem GPU			Com GPU		
Algoritmos	5.000	100.000	1.000.000	5.000	100.000	1.000.000
ABC	0,0034s	0,033	0,697s	<0,001s	<0,001s	<0,001s
Implementação 4	58s+ 1.39min	14.49min + 14.41min	2.54h+ 2.49h	46.11s + 26.23s	7.37min + 8.40min	1.19h+ 1.19 h
Implementação 5	54.35s + 54.14s	16.37min + 16.08min	-	42.00s + 26.11s	7.42min + 7.40min	1.20h+ 1.20h
Implementação 6	1.38min+ 55.42s	16.41min + 16.15min	-	26.23s + 26.67s	7.43min + 7.45min	1.24h+ 1.22h

Tabela 4.1: Comparação dos custos computacionais por implementação.

O algoritmo ABC apresenta os melhores tempos computacionais, porém ele já está bem implementado computacionalmente e serve apenas para questões de comparação com a distribuição real. É possível perceber também que os três algoritmos propostos têm tempos computacionais bem parecidos quando comparados entre si para mesmo tamanho de amostra e máquina, podendo discorrer que os métodos são igualmente custosos computacionalmente. Vale mencionar, também, a diferença de tempo no uso de GPUs, que têm tempo de processamento quase que a metade do tempo quando não utilizado para esses algoritmos em um mesmo tamanho de amostra.

Os gráficos mostrados abaixo representam as distribuições geradas a partir dos resultados dos algoritmos para tamanho de amostra igual a 1.000.000.

Para o treinamento do modelo de redes neurais foram utilizadas 100 épocas e máquina com GPU. Foram verificadas as métricas MSE e função de perda, sendo selecionada a configuração final dos parâmetros da rede neural a partir da época que teve o melhor desempenho. A Figura 4.5 mostra o comportamento dessas métricas por época.

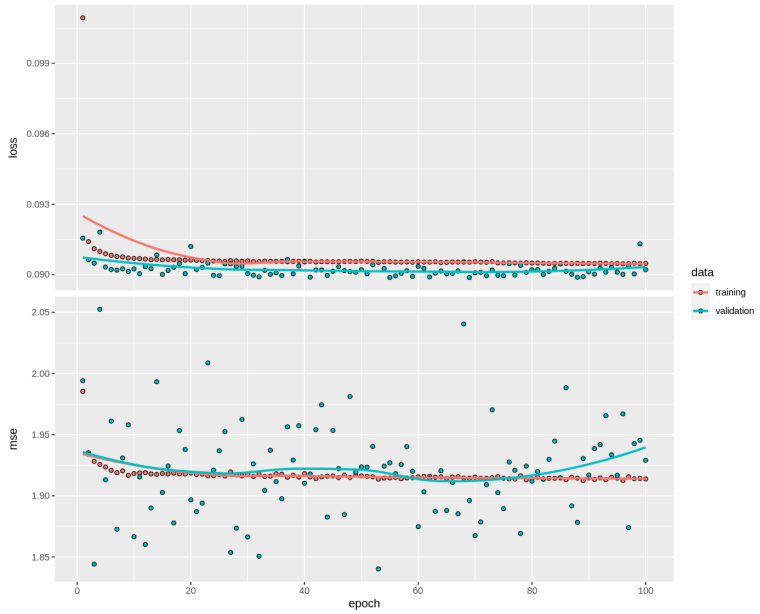
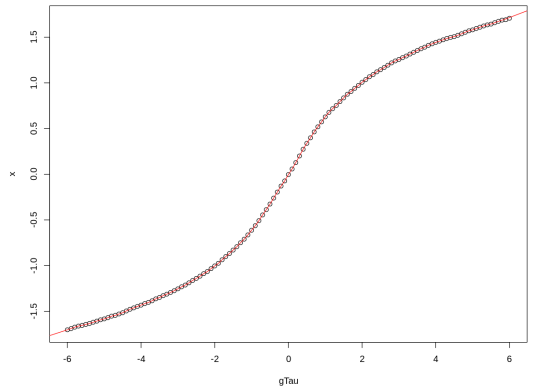


Figura 4.5: Histórico da função de perda e MSE durante as épocas treinadas

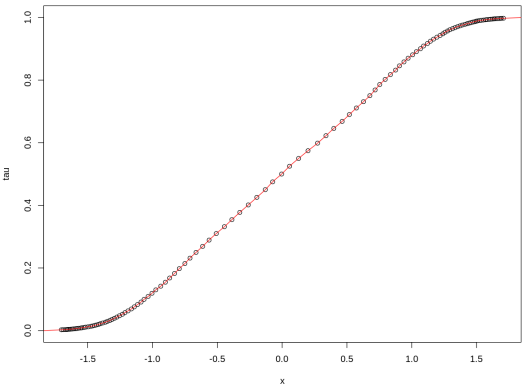
É possível perceber, pela Figura 4.5, que a partir da 20ª época o modelo atinge a menor função de perda no conjunto de treinamento, mantendo seu valor constante após essa época.

Após a estimação dos parâmetros do modelo de regressão quantílica via redes neurais, ajusta-se a predição de valores para obter a distribuição dos dados. O procedimento realizado requer um *grid* de valores aleatórios variando de 0 a 1 (que representam um valor do quantil), no qual serão aplicados a função *Logito* a esses valores. Após isso, prevê-se quais os valores da distribuição equivalem àquele quantil. Utiliza-se, por fim, o método Splines Monotônico para estimar uma curva contínua ao longo dos valores estimados, essa também chamada de distribuição acumulada. Aplicando a derivada, tem-se a distribuição de probabilidades e, com ela, é possível predizer pontos da distribuição.

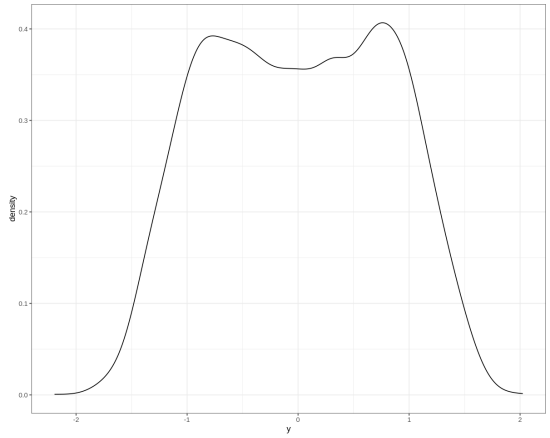
A Figura 4.6 ilustra como ocorre esse procedimento.



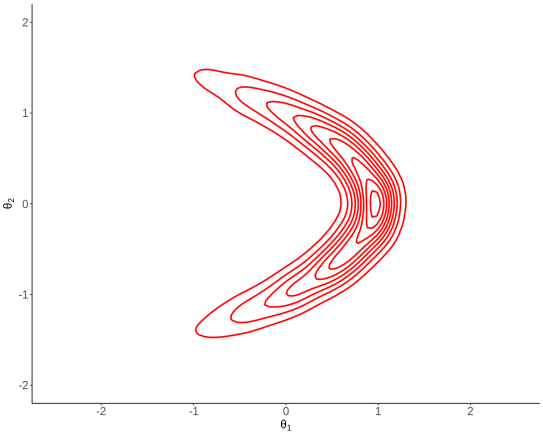
(a) Distribuição acumulada no espaço transformado dos pontos gerados para $\theta_1|y_{obs}$.



(b) Distribuição acumulada dos pontos gerados para $\theta_1|y_{obs}$ em cada quantil.



(c) Densidade de θ_1, após derivação da densidade acumulada.



(d) Distribuição conjunta de θ_1 e θ_2 após o procedimento

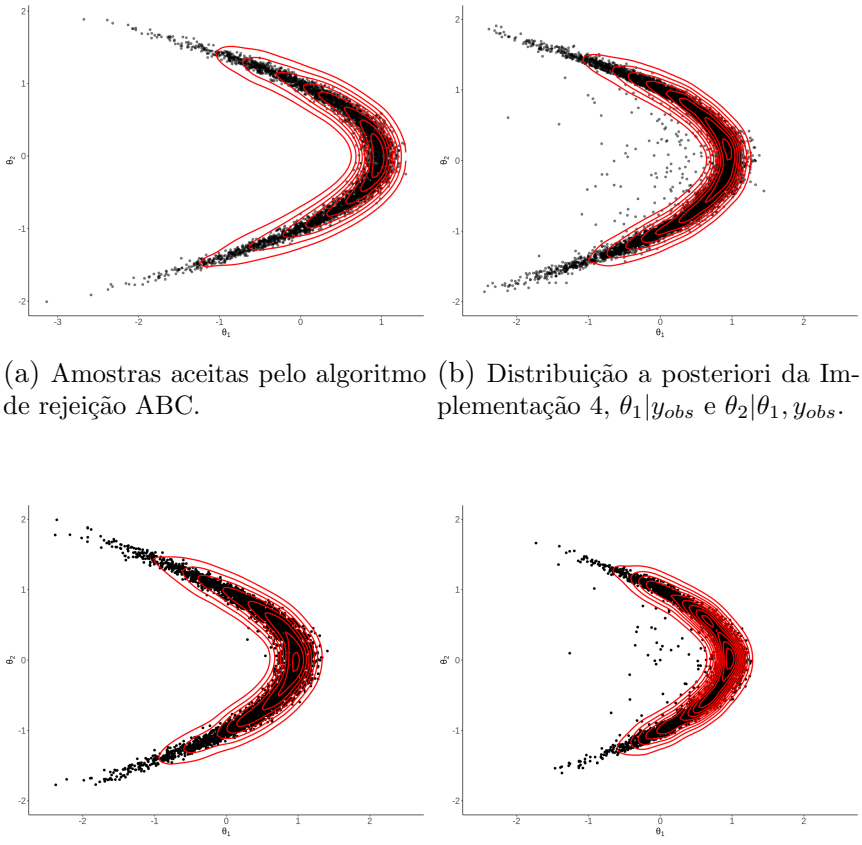
Figura 4.6: Gráficos dos procedimentos realizados para obtenção da distribuição a posteriori $\theta_1|y_{obs}$

O resultado final desse processo é uma distribuição acumulada de $\theta_1|y_{obs}$. Para encontrar a distribuição de probabilidades é necessário derivar y_{obs} em θ, e, por isso, utiliza-se a função Logito que é de fácil derivação.

Obtida a distribuição de probabilidades, o próximo passo do algoritmo é realizar as fatorações ou Amostrador de Gibbs, que são possíveis de ser computados através das distribuições de probabilidade dos parâmetros, resultanto, assim, na distribuição conjunta a posteriori.

Como mostrado pela Figura 4.6, o processo de modelagem da regressão quantílica via redes neurais e com ajuste do Splines Monotônico parece se ajustar bem aos dados. A estimação dos quantis conseguir absorver a característica dos dados e, com isso, obter uma distribuição conjunta a posteriori que tivesse comportamento parecido com a distribuição original.

Dado isso, partiu-se para as aplicações dos algoritmos. A Figura 4.7 ilustra o comportamento das amostras da distribuição a posteriori geradas pelo algoritmo ABC e pelas implementações 4 a 6.



(a) Amostras aceitas pelo algoritmo de rejeição ABC.

(b) Distribuição a posteriori da Implementação 4, $\theta_1|y_{obs}$ e $\theta_2|\theta_1, y_{obs}$.

(c) Distribuição a posteriori da Implementação 5, $\theta_2|y_{obs}$ e $\theta_1|\theta_2, y_{obs}$.

(d) Distribuição a posteriori da Implementação 6 - Gibbs.

Figura 4.7: Distribuições aproximadas da posteriori

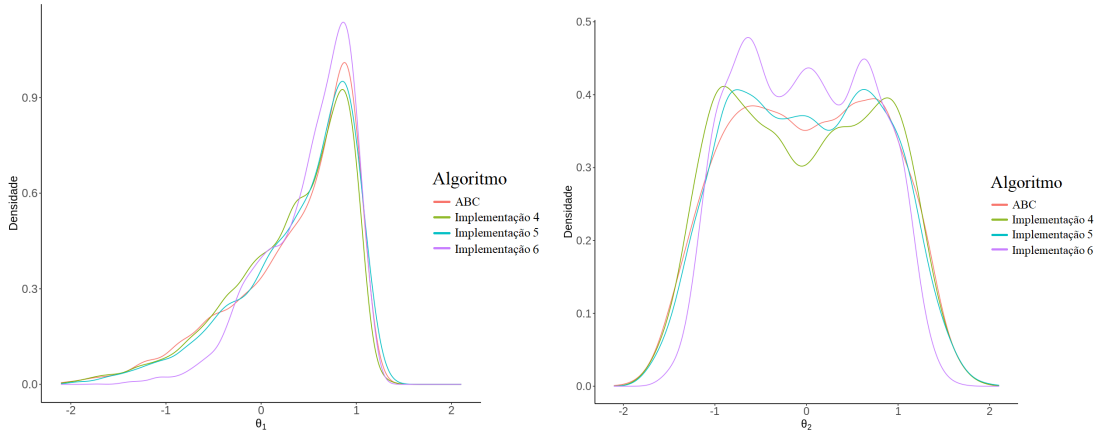
Considerando como base o algoritmo ABC, tem-se que a real distribuição do modelo

Twisted Normal tem o comportamento de concavidade, como mostrado na Figura 4.7 (a), tendo uma grande quantidade de pontos, denotada pelas curvas de nível, concentradas próximo de 1.

As implementações parecem ter comportamentos bem parecidos em relação a distribuição real, em que as curvas de nível indicam que a massa de pontos está proxima do comportamento da distribuição real. É possível observar, também, pontos fora do padrão para essa distribuição nas estimações dos algoritmos, isso pode inferir no desempenho do algoritmo. Em especial, cita-se a implementação 5, que tem como fatoração $(\theta_2|y_{obs})$ e $(\theta_1|\theta_2, y_{obs})$, por ser o algoritmo com menor quantidade de pontos fora do esperado para a distribuição real, dando indícios de que tenha a melhor perfomance entre os algoritmos.

Vale ressaltar que a quantidade de iterações utilizadas para as fatorações e no Amostrador de Gibbs foi de 5.000 unidades. Portanto, os gráficos da Figura 4.7 têm a mesma quantidade de pontos.

A Figura 4.8 ilustra a distribuição de densidades das amostras geradas da distribuição a posteriori por cada algoritmo para cada parâmetro. Observa-se que a linha correspondente ao algoritmo ABC pode ser tomada como referência para o parâmetro.



(a) Distribuições a posteriori de θ_1.

(b) Distribuições a posteriori de θ_2.

Figura 4.8: Distribuições a posteriori de θ_1 e θ_2.

Pelo Figura 4.8 (a) é possível perceber que as densidades de probabilidades têm comportamentos parecidos entre si.

Já na Figura 4.8 (b), os algoritmos também estão próximos entre si, mas, a configuração da distribuição da densidade parece variar entre -1 e 1.

Analisando o comportamento dos geradores de amostras da distribuição a posteriori (fatorações e Amostrador de Gibbs) aplicados, foram-se observados os comportamentos de autocorrelação e movimentação das amostras geradas, sendo esse analisado para as 50 primeiras observações. A Figura 4.9 ilustra bem esses comportamentos.

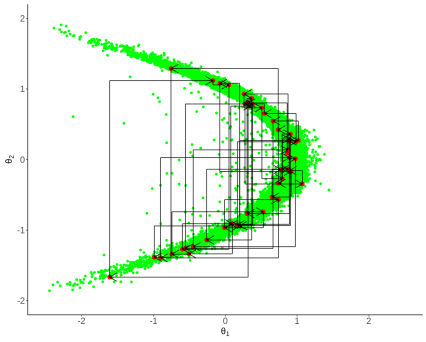
É possível perceber que o algoritmo ABC e os algoritmos que utilizam da fatoração (Implementação 4 e 5) têm comportamentos parecidos para os gráficos ACF (Função de autocorrelação) e de movimentação da cadeia. Esses comportamentos estão ligados entre si, pois é possível denotar que quanto mais independentes são os geradores de amostras da distribuição a posteriori, maior a variação que se tem ao gerar uma amostra consecutiva. E isso se faz valer para esses algoritmos, em que o ACF deles mostra que as amostras geradas são independentes entre si. Por consequência, seus gráficos de movimentação de cadeia mostram que há uma grande variação de uma amostra para a outra.

Já, para a Implementação 6, que utiliza o Amostrador de Gibbs para gerar amostras da distribuição a posteriori, tem-se, pelo gráfico ACF, que as amostras geradas apresentam autocorrelação serial, o que era de se esperar, pois é uma característica dos métodos MCMC.

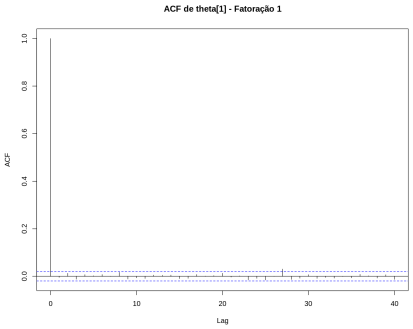
Vale ressaltar que o fato de selecionar as 50 primeiras amostras geradas serve apenas para efeito de visualização do caminho. Mas, esse padrão de variabilidade se repete para todas as amostras geradas, já que, para a maioria dos algoritmos, não há correlação serial.

Espera-se que as implementações que não apresentem autocorrelação serial tenham menor tempo de convergência para a distribuição a posteriori, necessitando, assim, menos iterações.

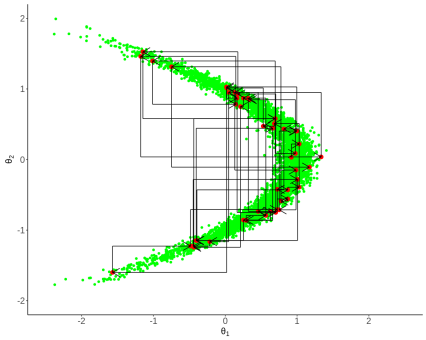
Como resultado final, pode-se inferir que a escolha da ordem da fatoração faz dife-



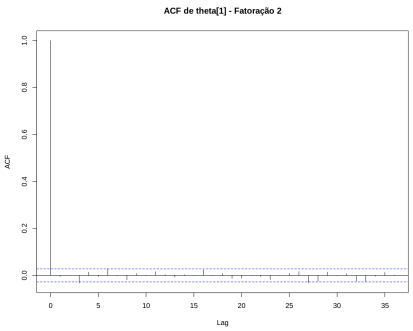
(a) ACF Implementação 4 - Fatoração $\theta_1|y_{obs}$ e $\theta_2|\theta_1, y_{obs}$.



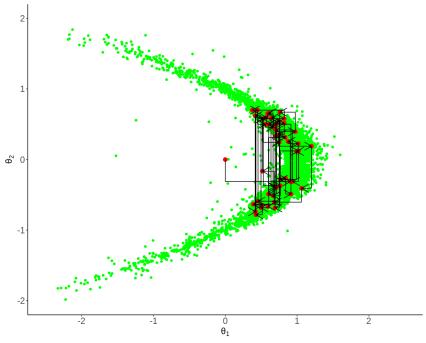
(b) Movimentação da Implementação 4 - Fatoração $\theta_1|y_{obs}$ e $\theta_2|\theta_1, y_{obs}$



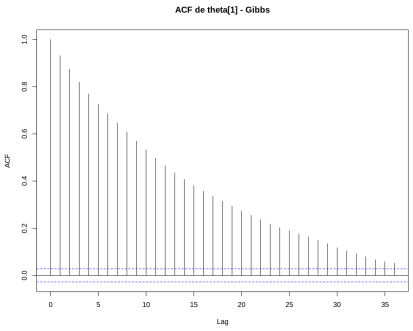
(c) ACF Implementação 5 - Fatoração $\theta_2|y_{obs}$ e $\theta_1|\theta_2, y_{obs}$.



(d) Movimentação da Implementação 5 - Fatoração $\theta_2|y_{obs}$ e $\theta_1|\theta_2, y_{obs}$.



(e) ACF Implementação 6 - Gibbs.



(f) Movimentação da Implementação 6 - Gibbs.

Figura 4.9: Mistura da cadeia e função de autocorrelação dos algoritmos.

rença na obtenção das distribuições a posterior, como visto pela comparação entre as implementações 4 e 5, na qual a primeira utiliza a ordem $\theta_1|y_{obs}$ e $\theta_2|\theta_1, y_{obs}$ e o segundo a ordem $\theta_2|y_{obs}$ e $\theta_1|\theta_2, y_{obs}$. Apesar disso, os algoritmos convergiram seus resultados, em sua maioria, com a distribuição exata do modelo.

Além disso, foram necessárias a geração de 1 milhão de amostras, o que levou a um certo custo computacional para os algoritmos que utilizam de redes neurais, e, também, para o algoritmo o ABC que precisou de muitas amostras para chegar a distribuição correta.

4.3 Mistura de Normais

Considerando agora um modelo de mistura gaussiano 2-dimensional apresentado por Rodrigues, Nott e Sisson (2019) na qual a função de verossimilhança é dada por:

$$p(\boldsymbol{s}|\boldsymbol{\theta}) = \sum_{b_1=0}^{1} \dots \sum_{b_D=0}^{1} \left[\prod_{i=1}^{D} w^{1-b_i}(1-w)^{1-b_i} \right] \phi_D(\boldsymbol{s}|\boldsymbol{\mu}(\boldsymbol{b}, \boldsymbol{\theta}), \boldsymbol{\Sigma}) \ ,$$

em que $\phi_p(\boldsymbol{x}|\boldsymbol{a}, \mathbf{B})$ denota a função de densidade da distribuição normal bivariada com média $\boldsymbol{a}$ e estrutura de covariâncias $\mathbf{B}$ avaliada em $\boldsymbol{x}$, $w \in [0, 1]$ é o peso da mistura, $\boldsymbol{\mu}(\boldsymbol{b}, \boldsymbol{\theta}) = ((1-2b_1)\theta_1, \dots, (1-2b_D)\theta_D)^\top$, $\boldsymbol{b} = (b_1, \dots, b_D)^\top$ com $b_i \in \{0, 1\}$ e $\boldsymbol{\Sigma} = [\Sigma_{ij}]$ tal que $\Sigma_{ii} = 1$ e $\Sigma_{ij} = \rho$ para $i \neq j$.

Os parâmetros utilizados para a distribuição foram, $D = 2$, $\boldsymbol{s}_{obs} = (5, 5)$ e $w = 0,3$ e $\rho = 0,7$. As estatísticas-resumo são dadas pela média da distribuição de cada variável em $\boldsymbol{X}$.

A diferença nos pontos das estatísticas-resumo observada com os valores do artigo original, de Rodrigues, Nott e Sisson (2019), se deve ao fato de estarmos interessados em avaliar se os algoritmos conseguem se mover entre as áreas de alta densidade da distribuição a posteriori. A distribuição a priori amostrada seguiu uma distribuição $U[-10, 10]$.

Os algoritmos utilizados foram:

i Implementação 3 - Regressão linear e fatorações da distribuição a posteriori, sendo o modelo com melhor comportamento apresentado na subseção 4.1;

ii Implementação 2 - Regressão quantílica e Amostrador de Gibbs Aproximado, como apresentado na subseção 4.2, e denotado por Santos (2021) na seção 2.6;

iii Implementação 4 - Regressão quantílica e fatorações da distribuição a posteriori, como apresentado na subseção 4.2 e proposto como inova]ao neste trabalho, denotado na seção 3.

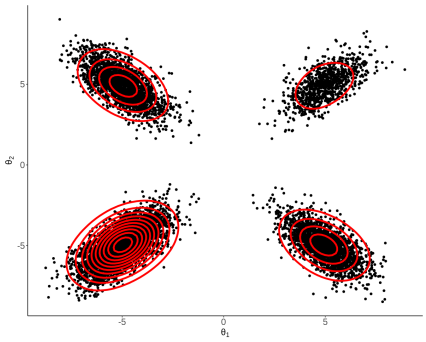
A arquitetura das redes neurais são as mesmas para ambos as implementações e estão descritas na seção 4.2. Assim como os tamanhos de amostras gerados (1.000.000) e taxa de aceitação (0,05).

A Figura 4.10 mostra as distribuições conjuntas geradas após o ajuste dos modelos e determinação das distribuições condicionais. Além disso, as densidades marginais de cada parâmetro obtido.

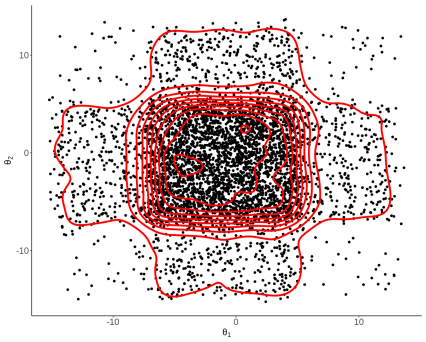
A Implementação 3 não obteve um bom ajuste se comparada a distribuição real dos dados, vide resultados do algoritmo ABC. Já as implementações 2 e 4 apresentaram comportamento mais próximo da distribuição exata, com alguns pontos fora das curvas de nível, sendo mais presente neste último algoritmo, indicando que, dentre os algoritmos utilizados nesse cenário, este último foi o que teve melhor desempenho.

É curioso notar que os pontos fora das curvas de nível na Figura 4.10 (c) parecem estar em um caminho diagonal, como se a cadeia estivesse se movimentando de uma região a outra oposta diagonalmente. O que ocorre diferentemente para o algoritmo da Figura 4.10 (d), em que os pontos parecem indicar que o movimento da cadeia vai de uma região a outra ortogonalmente, com pouca presença de ponto no centro do gráfico.

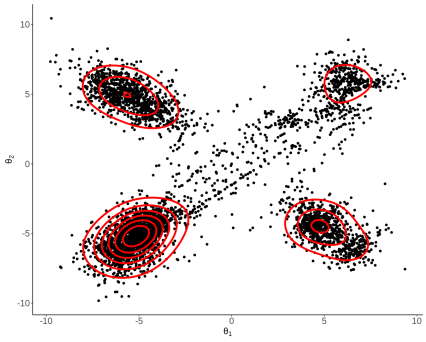
Pela avaliação das densidades dos parâmetros é possível perceber que a Implementação 3 se distancia bastante do comportamento dos outros algoritmos. Já Implementação 4



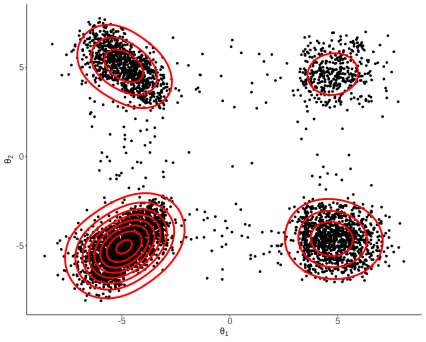
(a) Amostras aceitas pelo algoritmo de rejeição ABC.



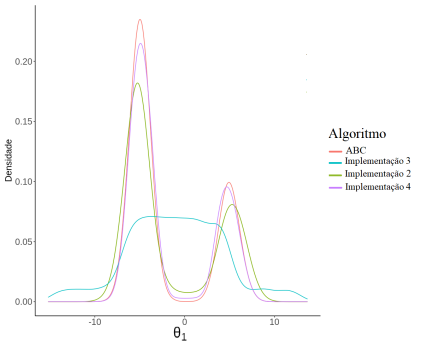
(b) Distribuição a posteriori da Implementação 3.



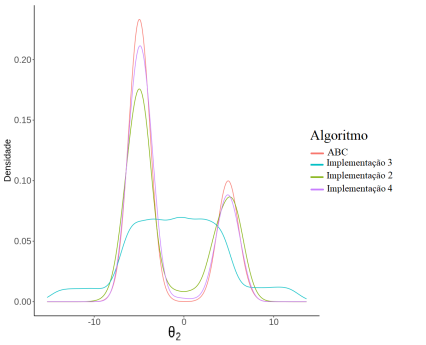
(c) Distribuição a posteriori da Implementação 2.



(d) Distribuição a posteriori da Implementação 4.



(e) Densidade marginal da distribuição a posteriori de θ_1 para cada implementação.



(f) Densidade marginal da distribuição a posteriori de θ_2 para cada implementação.

Figura 4.10: Distribuições conjuntas da distribuição a posteriori e densidades marginais.

tem sua curva de densidade mais próxima da curva do ABC, tanto para θ_1 quanto para θ_2.

A Figura 4.11 apresenta a mistura das cadeias geradas pelos algoritmos após o ajuste dos modelos de regressão, juntamente com os gráficos para avaliação da autocorrelação serial.

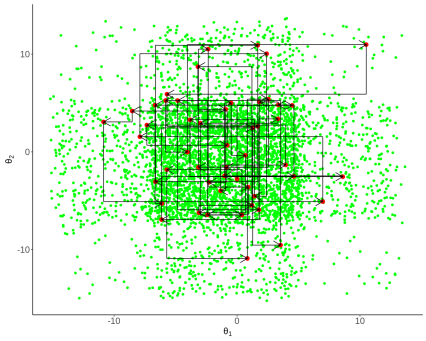
Como apresentado nos parágrafos anteriores, a Implementação 2 consegue caminhar suas iterações de forma diagonal, já a Implementação 4 não consegue. Isso mostra que, este último, acertou as outras regiões com mais exatidão e não precisou caminhar até chegar na região mais distante e menos massissa.

Vale observar que, na Figura 4.11 (c), nenhum ponto saiu de uma região e foi para sua oposta diagonalmente. O que pode ser um problema para o algoritmo, mostrando que precisa de pelo menos dois passos, para esse caso, para chegar a uma região oposta a sua.

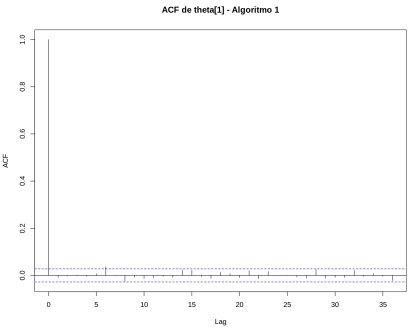
Pelos gráficos de autocorrelação, os três algoritmos apresentam comportamento satisfatório para essa análise, corroborando o que já foi dito anteriormente. Todos geram amostras da distribuição a posteriori independentes entre si, por isso não há uma correlação entre amostras consecutivas.

Isso se reflete nos gráficos de mistura, pois há uma relação nítida entre independência da distribuição a posteriori e os saltos que o algoritmo dá, mostrando maior flexibilidade, que é exigida para esse tipo de cenário de dados.

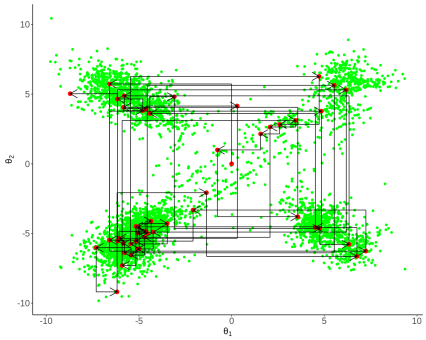
As figuras apresentadas nesta subseção corroboram com o bom comportamento da distribuição a posteriori gerada pela implementação 4 (Regressão quantílica via redes neurais, Splines Monotônico e geração da distribuição a posteriori a partir de aproximações por fatorações), proposta como inovação deste trabalho, havendo alta movimentação e boa convergência para a distribuição alvo.



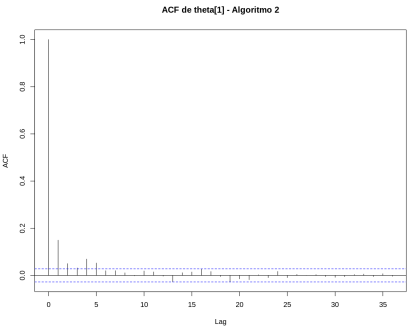
(a) Mistura da cadeia das 50 primeiras da Implementação 3



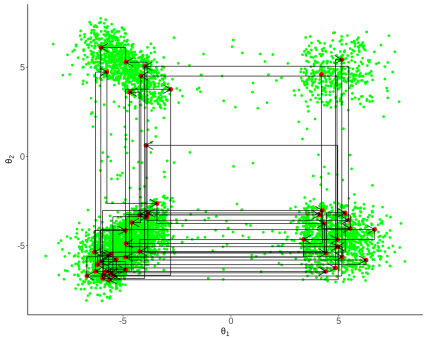
(b) Função de autocorrelação da Implementação 3



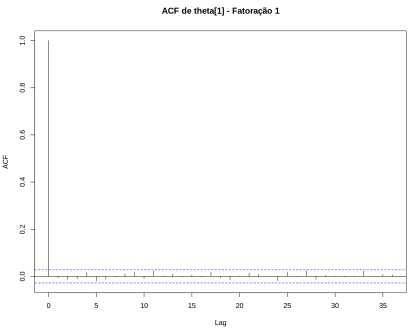
(c) Mistura da cadeia das 50 primeiras iterações da Implementação 2



(d) Função de autocorrelação da Implementação 2



(e) Mistura da cadeia das 50 primeiras iterações da Implementação 4



(f) Função de autocorrelação da Implementação 4

Figura 4.11: Mistura da cadeia e função de autocorrelação dos algoritmos

Capítulo 5

Conclusão

Este trabalho apresenta uma nova implementação eficiente de algoritmo ABC, utilizando a técnica de fatorações da distribuição a posteriori, apresentado como trabalhos futuros em Rodrigues, Nott e Sisson (2019) e juntamente com as inovações apresentadas em Santos (2021). Essa inovação foi aplicada nas implementação 3 da seção 4.1, implementações 4 e 5 da seção 4.2 e implementação 4 da seção 4.3.

O uso da regressão quantílica via redes neurais tem como finalidade gerar modelos mais flexíveis para os quantis, ao invés de modelos para as médias e variâncias, como ocorre em Rodrigues, Nott e Sisson (2019). Com isso, melhora-se as aproximações das densidades condicionais e diminui-se o erro na obtenção da distribuição a posteriori de interesse.

Foram realizadas comparações dessa implentação com as implementações desenvolvidas em Rodrigues, Nott e Sisson (2019) e em Santos (2021) para a obtenção dos parâmetros em três diferentes problemas: distribuição Normal Bivariada, Distribuição Twisted Gaussian e Distribuição Mistura de Normais. Em todas, a implementação proposta neste trabalho teve melhor comportamento se comparado aos algoritmos propostos em seus artigos.

Faz-se necessário, para trabalhos futuros, utilizar diferentes métricas para medir, com

exatidão, o erro gerado pelas amostras, já que o MSE em si não é o mais adequado para esse tipo de comparação. Medida de Kullback-Leibler parece ser a mais indicada para avaliar o quanto uma distribuição está próxima da outra.

Outra sugestão se dá pela substituição do método Splines Monotônico pela utilização da Regressão Isotônica, como alternativa para suavizar monotonicamente a curva da função de distribuição acumulada gerada pela Regressão Quantílica.

Algumas características dos algoritmos também foram verificadas, como a convergência, velocidade de mistura e independência das amostras geradas. Esses foram problemas citados em Rodrigues, Nott e Sisson (2019) e que podem impactar na eficiência do algoritmo. A implementação proposta neste trabalho se mostrou uma boa mistura na cadeia, resultando em uma convergência mais rápida nos valores, além disso, foi constatado que não houve a presença de autocorrelação serial. Esta também vista no algoritmo de Santos (2021), mas, graças ao uso da descorrelação prévia dos parâmetros.

A forma como a distribuição foi fatorada também foi estudada, em especial na seção 4.2, no qual a forma da fatoração afetou substancialmente a obtenção da distribuição a posteriori.

O tempo computacional também foi avaliado no cenário da distribuição Twisted Gaussian e não foram observadas diferenças significativas nos tempos entre os algoritmos estudados. Mas a diferença ocorre com o aumento do tamanho de amostras que são utilizadas. E muito desse tempo se deve ao fato de se estar utilizando redes neurais.

Referências

A. A. Strimmer, A. A. Kessy e A. A. Lewin e (2018). *Optimal whitening and decorrelation*. Vol. 72. The American Statistician Journal, pp. 309–314.

A. B. Lee, R. Izbicki e (2016). "Nonparametric Conditional Density Estimation in a High-Dimensional Regression Setting". *Journal of Computational and Graphical Statistics* 25, pp. 1297–1316.

A. C. P. L. F. Carvalho, K. Faceli e A. C. Lorena e J. Gama e (2011). *Inteligência Artificial: Uma Abordagem de Aprendizagem de Máquina*. LTC.

A. P. L. F. Carvalho, T. B. Ludemir A. P. Braga e (1998). *Fundamentos de Redes Neurais Artificiais*.

Abeywardana, S. (2018). "Deep Quantile Regression". *Towards Data Science*.

Academy, Data Science (s.d.). *Deep Learning Book*. URL: `http://www.deeplearningbook.com.br/`. Acesso em: 10/12/2020.

B. Murteira, C. D. Paulino e M. A. A. Turkman e (2003). *Estatística Bayesiana*. Fundação Calouste Gulbenkian.

C. S. Ong, M. P. Deisenroth e A. A. Faisal e (2020). *Mathematics for machine learning*. Cambridge University Press.

Cannon, A. J. (2018). "Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes." *Stoch Environ Res Risk Assess* 32, 3207–3225.

D. J. Balding, M. A. Beaumont e W. Zhang e (2002). *Approximate Bayesian computation in population genetics*. Vol. 162. Genetics, 2025–2035.

D. MacKay, I. Murray e Z. Ghahramani e (2006). *MCMC for doubly-intractable distributions*. Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI).

D. W. Wichern, R. A. Johnson e (1998). *Applied multivariate statistical analysis*. Prentice Hall International.

D.J. Balding, M.A. Nunes e (2010). "On optimal selection of summary statistics for approximate Bayesian computation." *Stat Appl Genet Mol Biol* 9.

E. I. George, G. Casella e (1992). *Explaining the Gibbs sampler*. Vol. 46. Am. Stat., 167–174.

G. Rech, MC Medeiros e T. Terasvita e (2002). *Building neural networks models for time series: a statistical approach*. Vol. 25, pp. 49–75.

H. A. Andrade, P. G. Kinas e (2010). *Introdução a análise Bayesiana (com R)*. maisQnada.

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

Haykin, S. (1999). *Neural Networks: A compreensive Foundation*. Prentice Hall.

– (2009). *Neural Networks and Learning Machines*. 3nd. Prentice Hall.

He, X. (1997). "Quantile Curves without Crossing". *The American Statistician* 51.2, pp. 186–192. DOI: 10.1080/00031305.1997.10473959.

I. Chronopoulos, A. Raftapostolos e G. Kapetanios (2021). "Deep Quantile Regression", Data Analytics for Finance and Macro Research Centre, ISSN 2516–5933.

Koenker, R. (2005). *Quantile Regression*. Econometric Society Monographs. Cambridge University Press. ISBN: 9780521608275,0521608279.

Koenker, R. e Bassett Jr., G. (1978). "Regression Quantiles", Econometrica, 46, 33–50.

M. A. Beaumont, E. Bazin e K. J. Dawson e (2010). *Likelihood-free inference of population structure and local adaptation in a Bayesian hierarchical model*. Vol. 185(2). Genetics, pp. 587–602.

M. A. Brubaker, I. Kobyzev e S. J. D. Prince e (2020). "Normalizing Flows: An Introduction and Review of Current Methods". *IEEE Transactions on Pattern Analysis and Machine Inteligence*. DOI: `arXiv:1908.09257`.

M. Evans, D. J. Nott e C. C. Drovandi e K. Mengersen e (2018). *Approximation of Bayesian Predictive p-Values with Regression ABC*. Vol. 13. Bayesian Anal.

M. G. B. Blum, K. Csillery e O. Francois e (2012). "abc: an R package for approximate Bayesian computation (ABC)". *Methods in Ecology and Evolution*.

M. Malohlava, E. LeDell e N. Gill e S. Aiello e A. Fu e A. Candel e C. Click e T. Kraljevic e T. Nykodym e P. Aboyoun e M. Kurka e (2019). *h2o: R Interface for 'H2O'. R*. Package Version: 3.26.0.2.

Martins, M. C. (2017). *Computação Bayesiana Aprosimada: aproximação em modelos de dinâmica populacional*. Tese de doutorado apresentada à Escola Superior de Agricultura Luiz de Queiroz, Piracicaba.

Meinshausen, N. (2006). "Quantile Regression Forests". *Journal of Machine Learning Research* 7, 983–999.

Mingot, A. A. (2005). *Análise de dados através de métodos de estatística multivariada: uma abordagem aplicada*. Vol. 162. Belo Horizonte: Editora UFMG, 2025–2035.

Moral, P. Del (1996). *Non Linear Filtering: Interacting Particle Solution*. Vol. 2(4). Markov Processes e Related Fields, pp. 555–580.

Morellato, S. A. (2014). *Inferência Estatística para Regressão Múltipla H-Splines*. Tese de Doutorado Instituto de Matemática da Universidade de Campinas.

O. Francois, M. G. B. Blum e (2010). *Non-linear regression models for approximate Bayesian computation*. Vol. 20. Statistics e Computing, 63–73.

P. Donnelly, S. Tavare e D.J. Balding e R.C. Griffiths e (1997). *Inferring Coalescence Times From DNA Sequence Data*. Vol. 145. Genetics, pp. 505–518.

R. A. Flauzino, I. N. Silva e D. H. Spatti e (2010). *Redes Neurais Artificiais para engenharia e ciências aplicadas*. São Paulo: Artliber.

R. E. Carlson, F. N. Fritsch e (1978). "Piecewise cubic interpolation methods". *UCRL-81230; CONF-781198-1*.

Rasteiro, L. R. (2017). *Regressão quantílica para dados censurados*. Dissertação de Mestrado do instituto de Matemática e Estatística da USP.

Rizzo, M. L. (2007). *Statistical computing with R*. Chapman e Hall/CRC.

Rodrigues, G. S. (2017). *New methods for infinite and high-dimensional approximate Bayesian computation*. Tese de Doutorado School of Mathematics e Statistics Faculty of Science, University of New South Wales (UNSW).

Rubin, D. B. (1984). *Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician*. Vol. 12. The Annals of Statistics, pp. 1151–1172.

S. A. Sisson, D. J. Nott e V. J. H. Ong e Y. Fan e (2018). *High-dimensional approximate Bayesian computation*. arXiv:1802.09725 [stat.CO].

S. A. Sisson, D. J. Nott e Y. Fan e L. Marshall e (2012). *Approximate Bayesian computation and Bayes linear analysis: Towards high-dimensional ABC*. Vol. 23(1). Journal of Computational e Graphical Statistics, 65–86.

S. A. Sisson, G. S. Rodrigues e D. J. Nott e (2020). *Likelihood-free approximate Gibbs sampling*. Stat Comput **30**, 1057–1073. DOI: `https://doi.org/10.1007/s11222-020-09933-x`.

S. A. Sisson, M. G. B. Blum e M. A. Nunes e D. Prangle e (2013). "A Comparative Review of Dimension Reduction Methods in Approximate Bayesian Computation". *Statist. Sci.* 28, pp. 189–208.

Santos, D. C. (2021). "Amostrador de Gibbs aproximado usando Computação Bayesiana Aproximada e regressão quantílica via redes neurais artificiais". *Dissertação de mestrado em estatística, Universidade de Brasília (Unb).*

Scott A. Sisson, Yanan Fan e Mark A. Beaumont (2019). *Handbook of Approximate Bayesian Computation*. 1 edition. Chapman e Hall/CRC, p. 679.

Silva, L. A. (2016). *Introdução à mineração de dados: com aplicações em R*. 1nd. Elsevier.

Taylor, J. W. (jul. de 2000). "A quantile regression neural network approach to estimating the conditional density of multiperiod returns". *Forecasting* 37, pp. 299–311.

W. Pitts, W. S. McCulloch e (1943). "A logical calculus of the ideas immanent in nervous activity". *The bulletin of mathematical biophysics* 5, 115–133.

W. Zhang, J. Fan e (1999). "Statistical Estimation in Varying Coefficient Models". *The Annals of Statistics* 27, 491–1518.