Universidade de Brasília

Instituto de Ciências Exatas

Departamento de Estatística

Dissertação de Mestrado

# Amostrador de Gibbs aproximado usando Computação Bayesiana Aproximada e regressão quantílica via redes neurais artificiais

**por**

**Débora Cristiane dos Santos**

**Orientador: Prof. Dr. Guilherme Souza Rodrigues**

Brasília, 18 de janeiro de 2021

# Amostrador de Gibbs aproximado usando Computação Bayesiana Aproximada e regressão quantílica via redes neurais artificiais

**por**

**Débora Cristiane dos Santos**

Dissertação apresentada ao Departamento de Estatística da Universidade de Brasília como requisito à obtenção do título de Mestre em Estatística.

Orientador: Prof. Dr. Guilherme Souza Rodrigues

Brasília, 18 de janeiro de 2021

**Texto aprovado por:**

Prof. Guilherme Souza Rodrigues
Orientador, EST/UnB

Prof. Leandro Tavares Correia
EST/UnB

Prof. Mariane Branco
IM/UFRJ

Deus, autor da vida, por fazer muito mais abundantemente além daquilo que pedimos ou pensamos.

Agradeço, primeiramente, a Deus que permitiu que eu chegasse até aqui. À minha família, pelo amor, incentivo e apoio incondicional.

Meus sinceros agradecimentos aos professores do PPGEST/UnB, em especial, ao meu orientador, Guilherme Souza Rodrigues, pelos constantes ensinamentos e pela capacidade de transferir conhecimento; aos integrantes do grupo de trabalho do qual o tema desta dissertação surgiu: Chris Drovandi, David J. Nott, Scott Sisson; aos colegas Adolfo, Alan, Jackson e Thaís que fizeram parte da minha formação e que irão continuar presentes em minha vida; aos gestores e equipe do Banco do Brasil, por me apoiarem nessa jornada; a todos que, direta ou indiretamente, fizeram parte da minha caminhada, o meu reconhecimento e gratidão!

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

*"A persistência é o caminho do êxito."* .

(Charles Chaplin)

# Resumo

Em muitos problemas de inferência Bayesiana em que os dados são complexos ou de alta dimensão, a distribuição marginal e a verossimilhança são analiticamente intratáveis ou difíceis de serem computadas, o que gera um obstáculo para obtenção da distribuição a posteriori. Na literatura Estatística existem diferentes métodos propostos para estimar a distribuição a posteriori nesses casos. Este trabalho teve como objetivo aprimorar, introduzindo duas inovações, o amostrador Gibbs-ABC proposto por Rodrigues, Nott e Sisson (2019), no qual, para contornar o problema da dimensionalidade, substitui-se, no algoritmo de Gibbs, as distribuições condicionais completas por aproximações obtidas por modelos de regressão ajustados sobre dados sintéticos. Propomos realizar a aproximação das condicionais completas acumuladas por meio do ajuste de regressão quantílica via redes neurais com correção *spline* monotônica, técnica que se destaca pela flexibilidade e capacidade de lidar com interações e não-linearidades nos dados. Além disso, na abordagem proposta é possível optar por utilizar uma estratégia de reparametrização/descorrelação dos parâmetros, antes de proceder com a estimação das condicionais completas. O desempenho do método proposto foi avaliado por meio de dois estudos de simulação, os quais apresentaram resultados satisfatórios.

Palavras-chave: Amostrador de Gibbs; Descorrelação; Inferência sem Verossimilhança; Redes Neurais; Regressão Quantílica.

# Abstract

In many Bayesian inference problems where the data is complex or large, the marginal distribution $p(\mathbf{X})$ and the likelihood $p(\mathbf{X}|\boldsymbol{\theta})$ are analytically intractable or difficult to be computed, which creates an obstacle to obtain the posterior distribution. In the Statistical literature, there are different methods proposed to estimate the posterior distribution in these cases. This work introduces two innovations to the Gibbs-ABC sampler proposed by Rodrigues, Nott e Sisson (2019), in which, to circumvent the dimensionality problem, in the Gibbs algorithm, the complete conditional distributions are replaced by approximations obtained by regression models, fitted over synthetic data. We propose to approximate the cumulative complete conditionals through the adjustment of quantile regression via neural networks with monotonic spline correction, a technique that stands out for its flexibility and ability to deal with the interections and non-linearities in the data. In addition, in our approach, it is possible to use a reparametrization/parameter decorrelation strategy, before proceeding with the estimation of the complete conditionals. The performance of the proposed method was analysed using two simulation studies, which shown satisfactory results.

Keywords: Decorrelation; Gibbs Sampler; Likelihood-free inference; Neural Networks; Quantile Regression.

# Sumário

# 1 Introdução

Em inferência Bayesiana, diferentemente da abordagem frequentista, os parâmetros, assim como os dados, são tratados como variáveis aleatórias, para os quais, uma distribuição de probabilidade denominada distribuição posteriori.

A posteriori é obtida por meio da combinação de uma distribuição a priori, que sintetiza o conhecimento prévio à observação dos dados, e da verossimilhança, que representa as informações oriundas dos dados.

Desta forma, a distribuição a posteriori, que reflete todo conhecimento atualizado sobre o parâmetro, é dada por $\pi(\boldsymbol{\theta}|\mathbf{X}) = p(\mathbf{X}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})/p(\mathbf{X})$ onde $\pi(\boldsymbol{\theta})$ representa a distribuição a priori, $p(\mathbf{X}|\boldsymbol{\theta})$ a verossimilhança e $p(\mathbf{X})$ a distribuição marginal que é uma constante de normalização.

Em muitos problemas de inferência Bayesiana, a distribuição marginal $p(\mathbf{X})$ é uma integral multidimensional que não pode ser analiticamente calculada. Além disso, se os dados são complexos, de alta dimensão ou com a função de probabilidade conhecida apenas por meio de um processo de geração de dados, é comum encontrar funções de verossimilhança $p(\mathbf{X}|\boldsymbol{\theta})$ analiticamente intratáveis ou difíceis de serem computadas (Murray, Ghahramani e MacKay, 2006), o que gera um obstáculo para obtenção da distribuição a posteriori, elemento chave de toda análise Bayesiana de dados (Kinas e Andrade, 2010). Nesses casos são necessários métodos numéricos para prosseguir com a inferência.

Na literatura Estatística existem diferentes métodos propostos para estimar a distribuição a posteriori, dentre estes métodos temos os que aproximam a distribuição a posteriori sem necessitar do cálculo de $p(\mathbf{X})$, como por exemplo, o Monte Carlo via Cadeia de Markov (MCMC), o Monte Carlo Seguencial (SMC) (DelMoral, 1996; Kinas e Andrade, 2010).

Nos casos em que a verossimilhança é intratável podemos citar os métodos de Inferência Indireta e Computação Bayesiana Aproximada (ABC) também conhecido como *likelihood-free method*.

O ABC é um método alternativo, que tem se destacado, para aproximação da posteriori, pois além de não precisar calcular a distribuição marginal $p(\mathbf{X})$ não é necessária a utilização de uma expressão explícita para a função de verossimilhança (Beaumont, Zhang e Balding, 2002; Murray, Ghahramani e MacKay, 2006; Rodrigues, 2017). Outros métodos podem ser encontrados em (Martins, 2017).

O ABC ganhou mais notoriedade após ser apresentado por Beaumont, Zhang e Balding

(2002) para tratar problemas complexos na área de genética que inviabilizam o cálculo da distribuição de verossimilhança de forma analítica. O método em vez de utilizar os dados completos, se baseia em medidas resumo dos dados (média, mediana, variância, dentre outras.) escolhidas pelo usuário, o que o torna computacionalmente viável. Há diversas formas de se implementar um algoritmo ABC, sendo sua versão mais simples baseada no algoritmo de rejeição. Esses métodos apresentam bom desempenho em problemas com baixa ou moderada dimensão, números de parâmetros inferiores a 50 (Rodrigues, Nott e Sisson, 2019). No entanto, sofrem com a questão de dimensionalidade (Blum e Francois, 2010), casos em que o método é ineficiente e tem dificuldade de gerar amostras aproximadas da posteriori (devido à baixa probabilidade de aceitação das amostras candidatas) (Nott et al., 2018b). De acordo com Rodrigues (2017), ainda está limitado a modelos de baixa dimensão.

Vários trabalhos propuseram soluções que buscam melhorar a eficiência do ABC. Beaumont, Zhang e Balding (2002) introduziram a ideia de reduzir o erro de aproximação por meio do ajuste de uma regressão linear local dos valores dos parâmetros simulados e o conjunto gerado das estatísticas de resumo. Bazin, Dawson e Beaumont (2010) exploraram o ajuste via ABC de modelos hierárquicos utilizando subconjuntos do vetor das estatísticas resumo por meio da fatoração da verossimilhança em componentes de baixa dimensão. Blum e Francois (2010) propuseram realizar o ajuste por regressão introduzido por Beaumont, Zhang e Balding (2002) usando modelos de regressão heterocedástica condicional não linear. Nott et al. (2018b) estimam distribuições marginais a posteriori univariadas de baixa dimensão, para subconjuntos do parâmetro de interesse e posteriormente reconstroem a distribuição posteriori conjunta com base nos resultados.

Rodrigues, Nott e Sisson (2019), apresentaram um amostrador de Gibbs aproximado, sem necessidade do cálculo da verossimilhança, no qual, para contornar o problema da dimensionalidade, substitui-se, no algoritmo de Gibbs, as distribuições condicionais completas (DCC) por aproximações obtidas por modelos de regressão.

Neste estudo foi dado foco a problemas de inferência estatística para modelos com verossimilhança $p(\boldsymbol{X}|\boldsymbol{\theta})$ não-calculável, tendo como objetivo aprimorar o amostrador Gibbs-ABC proposto por Rodrigues, Nott e Sisson (2019) de modo a acelerar a convergência e aumentar a velocidade de mistura (quão rápido uma cadeia consegue explorar completamente o suporte da distribuição alvo) visto que uma das desvantagens do método apresentada pelos autores é a mistura lenta na cadeia de Markov, em especial para modelos com fortes correlações à posteriori.

A Seção 2 descreve brevemente em que consiste o método de Monte Carlo, método ABC, descorrelação de variáveis, redes neurais, regressão quantílica, regressão quantílica via redes neurais e interpolação monotônica.

A Seção 3 aborda primeiramente o método para construir distribuições condicionais baseadas em regressão e o método para implementar o amostrador Gibbs aproximado sem verossimilhança apresentado por Rodrigues, Nott e Sisson (2019). Posteriormente, apresenta o método proposto neste trabalho para construir distribuições condicionais baseadas em regressão quantílica via redes neurais com correção *spline* monotônica com ou sem descorrelação dos parâmetros.

Na Seção 4, o desempenho do amostrador de Gibbs aproximado é avaliado para os métodos original e proposto em dois estudos de simulação: o primeiro estima a distribuição a posteriori de um modelo normal bivariado, no qual optou-se pelo processo em que há a etapa de descorrelação dos parâmetros; o segundo estima a distribuição a posteriori de um modelo de mistura gaussiana D-dimensional sem considerar a etapa de descorrelação. Em ambas as simulações utiliza-se regressão quantílica via redes neurais multicamadas para a estimação das condicionais completas.

Na Seção 5 são apresentadas as conclusões e considerações finais.

# 2 Background

## 2.1 Métodos de Monte Carlo

Os métodos de Monte Carlo podem se referir a qualquer algoritmo em análise numérica em que simulação computacional é usada. São empregados nas mais diferentes áreas, embora, para não fugir do escopo deste trabalho, será dada uma ideia geral do emprego desses métodos em inferência estatística, mais precisamente na estimação de parâmetros. Uma ideia mais geral do que são os métodos de Monte Carlo pode ser encontrada em Rizzo (2007).

Para melhor entendimento da ideia por trás dos métodos de Monte Carlo, considere que se deseja computar a seguinte integral

$$\int_{-\infty}^{\infty} g(x)dx. \tag{1}$$

Desde que as integrais estejam bem definidas, é possível fazer uso do seguinte artifício algébrico

$$\int_{-\infty}^{\infty} g(x)dx = \int_{-\infty}^{\infty} g(x)\frac{f(x)}{f(x)}dx = \int_{-\infty}^{\infty} h(x)f(x)dx = E(h(X)) = \theta, \quad (2)$$

Desta forma, o problema de computar a integral definida em (1) é transformado em um problema de valor esperado de uma função em relação a uma densidade arbitrária, $f(\cdot)$, que tenha o mesmo suporte do intervalo de integração. E, como é sabido, a Lei dos Grandes Números nos permite afirmar que a média amostral converge com probabilidade 1 para a média populacional $\theta$ (Rizzo, 2007). Ou seja, se é possível gerar uma amostra aleatória da densidade $f(\cdot)$, a média amostral de $h(X)$, definida na Equação 2, aproxima a integral definida em (1). A qualidade dessa aproximação será dada pelo tamanho da amostra.

## 2.2 Métodos de Monte Carlo via Cadeias de Markov

Segundo Rizzo (2007), muitas aplicações dos métodos de Monte Carlo via Cadeias de Markov surgem em Inferência Bayesiana, contexto em que as integrais utilizadas para o cálculo da posteriori são frequentemente difíceis de calcular por métodos numéricos, especialmente em altas dimensões. Os métodos de Monte Carlo via Cadeias de Markov fornecem uma forma para resolver de maneira satisfatória este tipo de problema de integração.

O maior desafio para empregar a metodologia abordada na seção 2.1 é gerar valores da distribuição $f(\cdot)$. Entretanto, se for possível construir uma Cadeia de Markov que tenha distribuição estacionária $f(\cdot)$, então pode-se executar a cadeia por um período longo de tempo de forma que a cadeia convirja (aproximadamente) para distribuição alvo.

Uma classe de algoritmos que permite a geração de valores conforme abordado é a classe de Algoritmos Metropolis-Hastings.

### 2.2.1 Amostrador Metropolis-Hastings (M-H)

Amplamente empregado nas mais diversas áreas, o M-H é um algoritmo mais geral que permite simular valores de uma Cadeia de Markov que tenha distribuição estacionária $f(\cdot)$ (Algoritmo 1). A ideia por trás deles é gerar uma Cadeia de Markov $\{\theta_t; t = 0, 1, 2, \ldots, N\}$ tal que a distribuição estacionária seja a distribuição alvo. Em resumo, o algoritmo deve especificar, para um determinado estado da cadeia, $\theta_t$ por exemplo, como gerar o próximo estado, $\theta_{t+1}$. Cabe ressaltar que em todos os algoritmos desta classe, existe um valor candidato $Y$, gerado a partir de uma distribuição de proposta $g(\cdot|\theta_t)$. Se o valor candidato for aceito, move-se a cadeia para

este estado. Caso contrário, a cadeia permanecerá em seu estado atual.

É importante notar que a distribuição geradora de candidatos pode, ou não, depender do estado corrente da cadeia, cabendo ressaltar que, caso a distribuição da proposta atenda às condições de regularidade (irredutibilidade e aperiodicidade), a cadeia convergirá para uma distribuição estacionária única $\pi$. O algoritmo é projetado para que a distribuição estacionária seja de fato a distribuição alvo, $f(\cdot)$. A demonstração desse fato pode ser encontrada em Rizzo (2007).

**Algoritmo 1:** Amostrador Metropolis-Hastings

**Entrada:**
Uma distribuição alvo $f(\cdot)$;
Uma distribuição geradora de candidatos $g(\cdot \mid \theta)$;
Um inteiro $N$ representando o número de amostras desejado;
Um valor inicial $\theta^{(0)}$;

*Simulação*

**para** $t = 1, \ldots, N$ **faça**
1. Gere $\theta^* \sim g(\cdot|\theta^{(t-1)})$;
2. Gere $U \sim U(0, 1)$;

**enquanto** $U > \frac{f(\theta^*)g(\theta^{(t-1)}|\theta^*)}{f(\theta^{(t-1)})g(\theta^* \mid \theta^{(t-1)})}$ **faça**
Repita (1) e (2);
**fim**
Faça $\theta^{(t)} = \theta^*$;
**fim**

**Saída:**
A cadeia $\{\theta^{(t)} \mid t = 1, \ldots, N\}$ contendo os valores gerados.

### 2.2.2 Amostrador de Gibbs

Caso especial do Amostrador Metropolis-Hastings, o Amostrador de Gibbs é frequentemente aplicado quando a distribuição alvo é uma distribuição multivariada. Supondo que todas as DCCs sejam conhecidas e fáceis de amostrar, o Gibbs atualiza um parâmetro (ou bloco de parâmetros) por vez, gerando candidatos pela amostragem das distribuições condicionais completas e, desta forma, todos os valores serão aceitos.

Para melhor entendimento do algoritmo, defina $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_d)$ um vetor aleatório em $\mathbb{R}^d$, e seja

$$\boldsymbol{\theta}_{-j} = (\theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_d),$$

o vetor removendo a componente $j$. O Algoritmo 2 descreve o Amostrador de Gibbs.

**Algoritmo 2:** Amostrador de Gibbs

**Entrada:**

Um inteiro $N$ representando o número de amostras desejado;

Um valor inicial $\boldsymbol{\theta}^{(0)}$;

`Simulação`

**para** $t = 1, 2, \ldots, N$ **faça**

Gere $\theta_1^{(t)} \sim f(\theta_1 |\, \theta_2^{(t-1)}, \theta_3^{(t-1)}, \ldots, \theta_d^{(t-1)})$

$\theta_2^{(t)} \sim f(\theta_2 |\, \theta_1^{t}, \theta_3^{(t-1)}, \ldots, \theta_d^{(t-1)})$

$\vdots$

$\theta_d^{(t)} \sim f(\theta_d |\, \theta_1^{t}, \theta_2^{t}, \ldots, \theta_{d-1}^{t})$;

**fim**

**Saída:**

A cadeia $\{\boldsymbol{\theta}^{(t)} \mid t = 1, \ldots, N\}$ contendo os valores gerados.

## 2.3 Computação Bayesiana Aproximada (ABC)

O método ABC aproxima a distribuição a posteriori dos parâmetros, por meio da simulação sob o modelo de interesse. Sendo que, no ABC a avaliação numérica da função de verossimilhança é substituída por uma avaliação da probabilidade do modelo ter produzido os dados observados, com base na simulação de conjuntos de dados *sintéticos* do modelo e na comparação deles com os dados observados.

A inferência por meio do ABC é realizada com base em estatísticas de resumo, ou seja, valores calculados para o conjunto de dados (média, mediana, variância, dentre outras.). A partir de uma priori definida para os parâmetros de interesse, são realizadas simulações do modelo, que são utilizadas para aproximar a verossimilhança sem avaliá-la explicitamente.

### 2.3.1 Algoritmo de Rejeição

Em sua formulação mais simples, conhecida como algoritmo de rejeição ABC, proposta inicialmente por Rubin (1984) e aprimorada por Tavare et al. (1997), é possível estimar a distribuição a posteriori para um parâmetro $\theta$ condicionado a um conjunto de dados observados $\boldsymbol{X}_{obs}$, sem conhecer a verossimilhança, o Algoritmo 3 traz sua implementação. Nesta abordagem trabalha-se com medidas resumo de baixa dimensão $\boldsymbol{s}_{obs} = \boldsymbol{S}(\boldsymbol{X}_{obs})$ informativas para $\theta$. A escolha ideal para essas medidas seria um conjunto de estatísticas suficientes para $\theta$, mas normalmente

não é possível garantir suficiência no contexto do ABC (Nott et al., 2018a). É importante salientar que se $\boldsymbol{S}$ não é suficiente há perda de informação e teremos uma aproximação sobre a qual não se tem garantias, mas se $\boldsymbol{S}$ é suficiente e o limite de tolerância $\delta \to 0$, o método é exato, no sentido de mirar a posteriori.

**Algoritmo 3:** Algoritmo de Rejeição ABC

**Entrada:**
Uma priori $\pi(\theta)$;
Um conjunto de dados observados $\mathbf{X}_{obs}$;
Um procedimento para gerar dados sob o modelo $p(\mathbf{X}|\theta)$;
Vetor de medidas resumo $\mathbf{S}()$ a serem computadas;
Limite de tolerância $\delta$;
Métrica de distância empregada $D()$;
Um inteiro $N$ representando o número de amostras desejado;

`Simulação`

**para** $t = 1, \ldots, N$ **faça**
1. Gere $\hat{\theta}^{(t)} \sim \pi(\theta)$;
2. Gere $\mathbf{X} \sim p(\mathbf{X} \mid \hat{\theta}^{(t)})$;

**enquanto** $D(\mathbf{S}(\mathbf{X}), \mathbf{S}(\mathbf{X}_{obs})) > \delta$ **faça**
Repita (1) e (2);
**fim**
**fim**

**Saída:**
A cadeia $\{\hat{\theta}^{(t)} \mid t = 1, \ldots, N\}$ contendo os valores gerados.

Cabe esclarecer que comparar $D(\mathbf{S}(\mathbf{X}), \mathbf{S}(\mathbf{X}_{obs})$ em vez de $D(\mathbf{X}, \mathbf{X}_{obs})$ e aceitar $\hat{\theta}$ em uma vizinhança de $\boldsymbol{s}_{obs}$ são concessões feitas para aumentar a probabilidade de aceitação das amostras. Para uma análise abrangente comparando o desempenho dos principais métodos de redução de dimensão propostas na literatura ABC, ver Blum et al., 2013 . O $D$ representa uma medida de distância (normalmente Euclidiana). Caso a magnitude dos parâmetros seja muito diferente, é preferível padronizar as variáveis para evitar que algum dos parâmetros domine a distância.

Apesar da simplicidade, o algorítimo é computacionalmente ineficiente, visto que é necessário gerar repetidamente conjuntos de dados completos do modelo para obter uma amostra aproximada da posteriori(Rodrigues, 2017). Valores altos para $\delta$ aumentam o erro sistemático gerado pelas aproximações (Tavare et al., 1997). A Figura 1 apresenta de forma simplificada o passo a passo do método.

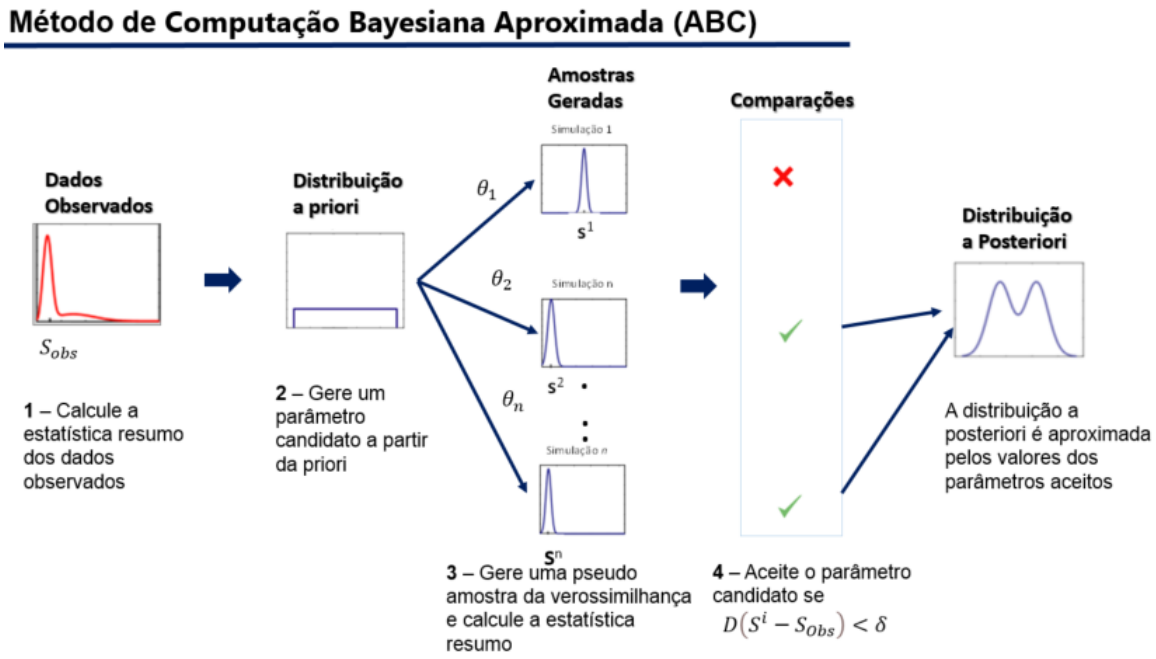



Figura 1: ABC - Adaptado do wikipedia.org

### 2.3.2 Amostragem por Importância

Esta é uma abordagem um pouco mais sofisticada do ABC em que é atribuído um peso relativo a importância de cada amostra a posteriori aproximada, em vez de simplesmente rejeitar ou aceitar um parâmetro candidato como no Algoritmo de Rejeição. A definição dos pesos se dá a partir de três componentes inter-relacionadas. Primeiro, uma função para calcular estatísticas resumos dos dados, $\boldsymbol{S}(\boldsymbol{X})$, é escolhida para reduzir a dimensão dos objetos. Nunes e Balding (2010) apresentam uma discussão detalhada sobre como selecionar a estatística resumo para obter resultados mais acurados na estimativa da posteriori via ABC.

Em segundo lugar, uma métrica de distância, $D(\boldsymbol{s}, \boldsymbol{s}_{obs}) = ||\boldsymbol{s} - \boldsymbol{s}_{obs}||$, é empregada para medir o grau de similaridade entre as estatísticas resumo de uma amostra sintética e as observadas $\boldsymbol{s}_{obs}$. Terceiro, uma função de ponderação, $K_h(d)$, que é controlada por um parâmetro de largura de banda $h$, define os pesos de importância de acordo com $D(\boldsymbol{s}, \boldsymbol{s}_{obs})$ - quanto maior a distância, menor o peso (Rodrigues, 2017).

Considere um vetor de parâmetros $D$-dimensional $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_D)^\top$ , com distribuição a priori associada $\pi(\boldsymbol{\theta})$, e verossimilhança $p(\mathbf{X}|\boldsymbol{\theta})$ intratável. A partir dos dados observados, $\boldsymbol{x}_{obs}$, pretende-se obter a distribuição a posteriori $\pi(\boldsymbol{\theta}|\boldsymbol{x}_{obs}) \propto p(\boldsymbol{x}_{obs}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$. Pode-se obter a

aproximação ABC por

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

em que $\boldsymbol{s} = \boldsymbol{S}(\boldsymbol{X})$ é um vetor de medidas resumo, $\boldsymbol{s}_{obs} = \boldsymbol{S}(\mathbf{X}_{obs})$ e $K_h(u) = K(u/h)/h$ é um kernel de suavização com parâmetro de banda $h > 0$. Um procedimento simples para coletar amostras de $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}_{obs})$ é dado no Algoritmo 4. Conforme dito na seção 2.3.1, se as medidas resumo $\boldsymbol{s}$ são suficientes, o erro de aproximação pode ser considerado pequeno, ou seja a banda $h \to 0$, neste caso $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}_{obs})$ convergirá para a distribuição exata a posteriori $\pi(\boldsymbol{\theta}|\boldsymbol{x}_{obs})$.

**Algoritmo 4:** Amostragem por importância ABC

**Entrada:**

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

*Início*

**para** $i = 1, 2, \ldots, N$ **faça**

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

**fim**

**Saída:**

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

### 2.3.3 Exemplo Introdutório

Para auxiliar na compreensão da técnica, replicou-se o exemplo introdutório de estimativa via ABC apresentado por Rodrigues (2017), utilizando o Algoritmo 4. Com foco principal em seus aspectos operacionais, explorou-se em um nível gráfico e intuitivo - um exemplo de simulação. Para tal, considerou-se um modelo simples caracterizado por uma sequência de eventos estocásticos dependentes. Precisamente, cada observação, $x_i,\ i = 1, \cdots, N$, é obtida por (a) uma amostra de uma distribuição normal com média $\theta$ e variância 1; e (b) se o valor obtido em (a) for negativo, ele será multiplicando por -1 com probabilidade $p = 0,5$. A característica fundamental é que a ação, ou a falta dela, na etapa (b) é condicional ao que aconteceu em (a).

Este modelo pode ser descrito pelas seguintes equações:

$$x_i = \begin{cases} |y_i|, \text{ com probabilidade } p = 0,5 \\ y_i, \text{ com probabilidade } 1 - p = 0,5 \end{cases}$$

$$y_i \sim N(\theta, 1).$$

Atribui-se uma priori gaussiana padrão para $\theta$, que completa o processo de geração de dados (priori preditiva). Condicionado em $\theta = -1$, que tomamos como o valor "verdadeiro" do parâmetro, simulamos 50 amostras do modelo para atuar como o conjunto de dados "observado".

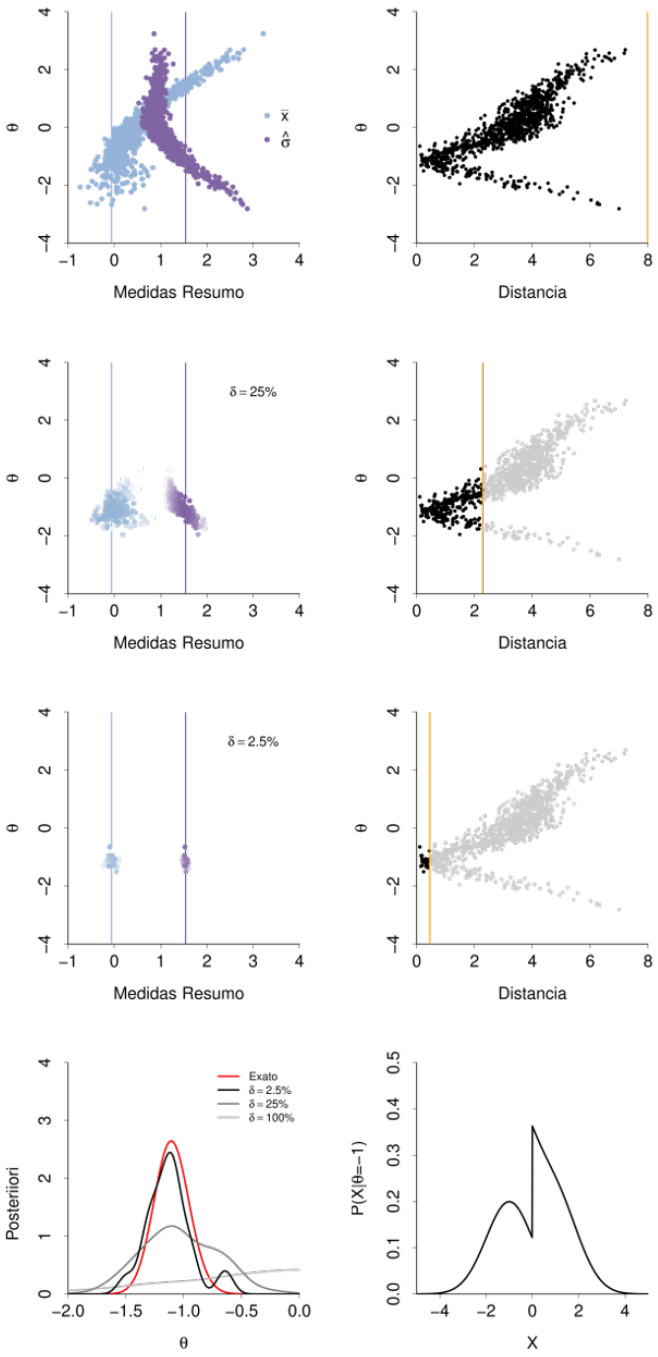
A função densidade $p(\mathbf{X}|\theta = -1)$ é representada no canto inferior direito da Figura 2. Neste exemplo a verossimilhança pode ser facilmente avaliada, mas em construções mais elaboradas, pode ser intratável. Os dados univariados são resumidos por sua média amostral, $\bar{x}$, e o desvio padrão, $\hat{\sigma}$. A distância Euclidiana e o kernel de Epanechnikov foram escolhidos como a medida de distância e a função de ponderação, respectivamente. Esta função tem a propriedade de atribuir peso zero para amostras posteriores cuja distância entre o conjunto de dados sintético e a estatística de resumo observada é maior do que h. Ou sena, as amostras insatisfatórias são descartadas, aumentando a eficiência computacional (Fan e Zhang, 1999). O Algoritmo 4 foi então usado para gerar $N = 1000$ amostras aproximadas da distribuição a posteriori ABC para uma banda de tolerância $h$.

O gráfico superior esquerdo na Figura 2 mostra os objetos gerados pelos Passos 1 a 3 do Algoritmo 3. Cada parâmetro $\theta^{(1)}, \cdots, \theta^{(N)}$ está associado a dois pontos no gráfico - um representando a média e o outro o desvio padrão da amostra sintética correspondente. A relação entre o parâmetro de interesse e as medidas resumo forma um padrão interessante, quando $\theta$ é grande, a probabilidade de gerar uma amostra negativa $y$ torna-se desprezível, de modo que a função de densidade torna-se efetivamente uma única distribuição normal centrada em $\theta$. Nesta região, a média amostral é suficiente para $\theta$, enquanto o desvio padrão se torna não informativo - observe o agrupamento formado em torno do verdadeiro desvio padrão $\sigma = 1$. Do outro lado do eixo de coordenadas, quando $\theta$ é, digamos, menor que -3, então cada ponto de dados gerado é refletido com probabilidade 0,5, transformando a função densidade em uma mistura de duas normais com médias opostas. Nesse caso, $\bar{x}$ estima $E(X|\theta) \approx 0$, sem nenhuma informação útil sobre $\theta$. O desvio padrão, entretanto, assume uma relação linear negativa com $\theta$.

As linhas verticais representam as medidas resumo "observadas". Vemos que o modelo só pode reproduzir $\hat{\sigma} \approx 1,5$ se $\theta$ estiver em uma vizinhança de -1. Os gráficos do lado direito exploram

a relação entre $\theta$ e as distâncias Euclidianas calculadas $||\mathbf{s}^{(i)} - \mathbf{s}_{obs}||$. Novamente, os padrões observados sugerem que a distância só pode ser pequena se $\theta \approx -1$.

A etapa 4 do Algoritmo 4 é ilustrada pelos gráficos da Figura 2. Primeiro, definimos $h$ para que 25% dos parâmetros candidatos recebam pesos positivos. Como a maioria das amostras originais foi descartada, as nuvens de pontos agora se agrupam mais próximas às linhas verticais. Conforme $h$ é reduzido, as amostras se aproximam ainda mais do espaço sobre o qual a posteriori exata é apoiada. Por outro lado, o número de amostras aceitas também diminui, o que leva a um aumento do erro de Monte Carlo. Essa compensação, que é uma característica bem conhecida dos métodos ABC, é observada no gráfico inferior esquerdo da Figura 2. Conforme ilustrado em nosso exemplo introdutório, definir um sistema de ponderação apropriado é fundamental para aliviar o erro de aproximação ABC.




Fonte: Rodrigues (2017)

Figura 2: Os gráficos do canto superior esquerdo mostram as medidas resumo sintéticas (pontos) e observadas (linhas verticais) para diferentes taxas de aceitação $\delta$. A intensidade da cor reflete os pesos de importância (calculados a partir da função kernel de Epanechnikov), com o branco liso representando peso zero. Os gráficos correspondentes no lado direito apresentam as distâncias Euclidianas entre as estatísticas resumos sintéticas e observadas. As amostras em cinza claro receberam peso zero (e, portanto, rejeitadas). A linha laranja representa a tolerância $h$ induzida por $\delta$. O gráfico inferior esquerdo compara os posteriores exatos (vermelho) e ABC aproximados (em cinza). O gráfico inferior direito mostra a função de densidade para dados, condicionada ao "verdadeiro" valor do parâmetro $\theta = -1$.

### 2.3.4 Ajuste do ABC por Regressão

O parâmetro de largura de banda $h$ é um parâmetro de distância e controla a influência dos dados vizinhos em um Kernel. Para mitigar o efeito de $h > 0$, métodos de pós-processamento de ajuste de regressão com base nas amostras ponderadas, segundo a proximidade com as distribuições marginais intratáveis, são comumente usados (Veja Beaumont, Zhang e Balding (2002); Blum e Francois (2010); Rodrigues, Nott e Sisson (2019)).

O método ABC com ajuste por regressão é uma versão mais sofisticada para estimar a posteriori $\pi(\boldsymbol{\theta}|\mathbf{X})$, inicialmente proposto por Beaumont, Zhang e Balding (2002) e posteriormente por Blum e Francois (2010), com objetivo de melhorar a qualidade da aproximação via ABC e deixar o algoritmo computacionalmente mais eficiente.

A ideia central do método é, a partir de amostras ponderadas $\{(\boldsymbol{\theta}^{(i)}, \boldsymbol{s}^{(i)}, w^{(i)})\}_{i=1}^{N}$, para $w^{(i)} > 0$, ajustar modelos da forma $\theta_d|\boldsymbol{S} \sim f(\theta_d|\boldsymbol{\beta}_d, \boldsymbol{S})$, $d = 1, \ldots, D$, na vizinhança de $\boldsymbol{s}_{obs}$. Um exemplo é dado em Beaumont, Zhang e Balding (2002) em que é ajustado um modelo de regressão local, método não paramétrico que utiliza suavização para ajustar curvas e superfícies aos dados, a partir da equação

$$\theta_d^{(i)} = \alpha_d + \boldsymbol{\beta}_d^{\top}(\boldsymbol{s}^{(i)} - \boldsymbol{s}_{\text{obs}}) + \epsilon_d^{(i)},$$

para $i = 1, \ldots, N$ e $d = 1, \ldots, D$, em que $\alpha_d \in \mathbb{R}$, $\boldsymbol{\beta}_d \in \mathbb{R}^q$, $q$ é a dimensão do vetor de medidas resumo $\boldsymbol{s}$, e $\epsilon_d^{(i)} \sim N(0, \sigma_d^2)$. Definimos como $\boldsymbol{\beta}_d^{+} = (\alpha_d, \boldsymbol{\beta}_d, \sigma_d^2)^{\top}$ o vetor dos parâmetros desconhecidos do modelo $d$.

Posteriormente, para reduzir a discrepância entre $\boldsymbol{s}^{(i)}$ e $\boldsymbol{s}_{\text{obs}}$, procede-se com uma modificação do $\theta_d^{(i)}$ da forma $\theta_d^{*(i)} = \hat{\boldsymbol{\beta}}_d^{\top}\boldsymbol{s}_{\text{obs}} + (\theta_d^{(i)} - \hat{\boldsymbol{\beta}}_d^{\top}\boldsymbol{s}^{(i)})$, em que $\hat{\boldsymbol{\beta}}_d$ é uma estimativa de $\boldsymbol{\beta}_d$.

A modelagem pode ser realizada com diferentes abordagens de regressão. O pacote `abc` do R, por exemplo, implementa diferentes modelos de regressão para ajuste, incluindo regressão linear, regressão ridge e redes neurais (Csillery, Francois e Blum, 2012).

Blum e Francois (2010) estenderam o modelo linear local de duas maneiras interessantes. Primeiramente, propuseram uma construção heterocedástica para explicar explicitamente as variâncias de erro desiguais.

Posteriormente, somando-se ao esforço de tornar o ajuste de regressão mais robusto e adequado para uma classe mais ampla de modelos intratáveis sugeriram estimar as densidades condicionais completas a partir do ajuste de modelos via redes neurais, em que são modelados

tanto parâmetro de posição $\boldsymbol{\mu}$ quanto de escala $\boldsymbol{\sigma}$ em segundo estágio, com base nos resíduos do primeiro. Ao amostrar dos resíduos assume-se que a distribuição é a mesma em todo espaço.

Nesta abordagem, a média condicional $\mathbb{E}(\theta_d|\ldots)$ é uma função não linear das covariáveis e a variância condicional $\mathbb{V}(\theta_d|\ldots)$ não é constante em todo o espaço das covariáveis. Desta forma, a distribuição condicional é aproximada por:

$$\theta_d|(\boldsymbol{S}, \boldsymbol{\theta}_{-d}) \quad = \quad m(g_{\theta_d}(\cdot)) + \sigma(g_{\theta_d}(\cdot)) \times \zeta,$$

em que $\zeta$ é uma variável aleatória com média zero e variância fixa. A esperança condicional é estimada em $\hat{m}(g_{\theta_d}(\cdot))$ por meio de uma rede neural. O termo de variância $\sigma(g_{\theta_d}(\cdot))$ é similarmente estimado por uma rede neural ajustada sobre os resíduos quadrados $r^2 = (\theta_d - \hat{m}(g_{\theta_d}(\cdot)))^2$.

Uma amostra aproximada da distribuição condicional completa é então dada por

$$\theta_d^* = \hat{m}(g_{\theta_d}(\cdot)) + \hat{\sigma}(g_{\theta_d}(\cdot)) \times \frac{{\theta_d}^{(i)} - \hat{m}^{(i)}}{\hat{\sigma}^{(i)}},$$

em que, $i$ é aleatoriamente selecionado $1, \ldots, N$.

## 2.4 Descorrelação de Variáveis

Gamerman e Lopes (2006) afirmam que uma boa parametrização é útil para melhorar a mistura da cadeia e acelerar a convergência. Segundo Paulino, Turkman e Murteira (2003)

> *"A velocidade de convergência para distribuição alvo da cadeia de Markov gerada de modo a obter uma amostra dela depende estreitamente do grau de correlação entre os elementos do vetor aleatório que tal distribuição impõe. Quando os componentes a posteriori são independentes ou fracamente correlacionados, o produto do amostrador Gibbs move-se mais livremente em torno de uma gama mais ampla das distribuições condicionais completas."*

A melhor maneira de se estudar a descorrelação é entender o que ocorre no caso mais simples, com apenas duas variáveis, $X$ e $Y$. Elas são ditas descorrelacionadas quando:

$$E(XY) = E(X)E(Y)$$

Assim, quando duas variáveis são linearmente descorrelacionadas, o valor esperado do produto entre elas é igual ao produto dos valores esperados, e portanto, sua covariância será igual a zero. Para os casos em que a média das variáveis é nula a condição passa a ser que

$$E(XY) = 0$$

Ou seja, uma condição de ortogonalidade.

Neste ponto, é possível estabelecer uma ligação entre descorrelação e a Análise de Componentes Principais (PCA), uma vez que, durante o processo de obtenção das componentes é feita a imposição de que elas sejam ortogonais entre si. Assim, é possível usar as componentes principais como forma de se obter novas variáveis descorrelacionadas que tenham pouca perda de informação em relação às variáveis originais.

### 2.4.1 Análise de Componentes Principais para Descorrelação de Variáveis

Kessy, Lewin e Strimmer (2018) analisaram diversos métodos que podem ser utilizados para descorrelação/*whitining* de variáveis, dentre os quais a PCA apresentou melhores resultados, o que motiva o seu uso neste estudo.

A Análise de Componentes Principais possui diversas aplicações, como por exemplo, a redução de dimensionalidade, a estimação de fatores na análise fatorial e a eliminação de multicolinearidade em uma regressão. A ideia básica por trás da PCA é, a partir de um conjunto de variáveis correlacionadas, buscar, por meio de combinações lineares dessas, variáveis latentes não correlacionadas (linearmente) entre si. Isto pode ser feito por meio da diagonalização da estrutura de covariâncias das variáveis sob estudo (Johnson e Wichern, 1998); (Mingot, 2005).

Do ponto de vista algébrico, a diagonalização pode ser realizada por meio da decomposição espectral da matriz $\Sigma$ em autovalores e autovetores, que apresentam propriedades relevantes. A primeira delas é a de que os autovetores gerados pela decomposição são uma base ortonormal - os autovetores são unitários e perpendiculares entre si. Outra propriedade é a de que os autovalores obtidos coincidem com a variância das componentes.

Em geral, para o estudo de um fenômeno com $p$ variáveis correlacionadas, obtém-se $p$ variáveis latentes que devem ser mutuamente não correlacionadas. É usual ordenar essas variáveis em função de suas variâncias. Deste modo, a primeira variável latente é definida pela combinação linear de maior variância, a segunda por aquela que possuir a segunda maior variância e

assim sucessivamente, até a $p$-ésima variável, a de menor variância.

Cabe notar que esta é uma técnica, a princípio, puramente matemática, no sentido de que nenhuma distribuição de probabilidade é associada às variáveis sob estudo. Além disso, ela só depende da estrutura de covariância (ou correlação linear) do conjunto de variáveis sob estudo.

Sendo um pouco mais formal, seja $\boldsymbol{X}^\top = [X_1, \ldots, X_p]$ o vetor aleatório com matriz de covariância associada $\boldsymbol{\Sigma}$.

Se as combinações lineares de $\boldsymbol{X}^\top$ forem definidas como

$$Y_i = \boldsymbol{a}_i^\top \boldsymbol{X} = a_{i1}X_1 + a_{i2}X_2 + \ldots + a_{ip}X_p, \quad i = 1, 2, \ldots, p,$$

decorre que (veja Johnson e Wichern, 1998 p. 400),

$$Var(Y_i) = \boldsymbol{a}_i^\top \boldsymbol{\Sigma} \boldsymbol{a}_i, \quad i = 1, 2, \ldots, p$$
$$Cov(Y_i, Y_k) = \boldsymbol{a}_i^\top \boldsymbol{\Sigma} \boldsymbol{a}_k, \quad i, k = 1, 2, \ldots, p.$$

Impondo às combinações lineares acima que elas sejam ortogonais e que as respectivas variâncias sejam as máximas possíveis, a elas dá se o nome de componentes principais.

A $i$-ésima componente principal $Y_i$ é a combinação linear de máxima variabilidade restrita a $\boldsymbol{a}_i^\top \boldsymbol{a}_i = 1$. Esta restrição se deve ao fato de que ao se maximizar a variância de $Y_i$, ou forma quadrática $\boldsymbol{a}_i^\top \boldsymbol{\Sigma} \boldsymbol{a}_i$, seu máximo pode ser aumentado multiplicado $\boldsymbol{a}_i$ por uma constante. Desta forma, ela faz-se necessária.

Um fato interessante que decorre após a maximização da variância de $Y_i$ segue no seguinte resultado.

**Resultado 1.1** Seja $\boldsymbol{\Sigma}$ a matriz de covariâncias associada ao vetor aleatório $\boldsymbol{X}^\top = [X_1, \ldots, X_p]$. Se $\boldsymbol{\Sigma}$ tem os pares de autovalores e autovetores $(\lambda_1, \boldsymbol{e}_1), (\lambda_2, \boldsymbol{e}_2), \ldots, (\lambda_p, \boldsymbol{e}_p)$, então a i-ésima componente principal é dada por

$$Y_i = \boldsymbol{e}_i^\top \boldsymbol{X} = e_{i1}X_1 + e_{i2}X_2 + \ldots + e_{ip}X_p.$$

Com estas escolhas,

$$Var(Y_i) = \boldsymbol{e}_i^\top \boldsymbol{\Sigma} \boldsymbol{e}_i = \lambda_i, \quad i = 1, 2, \ldots, p$$
$$Cov(Y_i, Y_k) = \boldsymbol{e}_i^\top \boldsymbol{\Sigma} \boldsymbol{e}_k = 0, \quad i \neq k.$$

Se alguns $\lambda_i$ forem iguais, as escolhas dos correspondentes coeficientes dos vetores $\boldsymbol{e}_i$, e consequentemente $Y_i$, não serão únicas. Tem-se portanto que as componentes principais são não correlacionadas e têm variâncias iguais aos autovalores de $\boldsymbol{\Sigma}$.

O PCA convencional trata todos os dados uniformemente e não explora qualquer conhecimento da qualidade relativa de cada amostra. Se a variação do ruído de cada amostra for conhecida, uma abordagem mais natural é dar mais peso para amostras menos ruidosas, ou seja, optar por utilizar o PCA ponderado, que pode ser obtido a partir dos passos apresentados no Algoritmo 5.

**Algoritmo 5:** PCA Ponderada

**Entrada:**
Um conjunto de valores $\boldsymbol{\theta}$;
Uma matriz diagonal de pesos $\mathbf{W}$;
*Início*
Obtenha o vetor de médias $\boldsymbol{\mu}$ de $\boldsymbol{\theta}$;
Obtenha a estrutura de covariâncias ponderada $\boldsymbol{\Sigma} = \frac{1}{\sum_i w_{ii}}(\boldsymbol{\theta} - \boldsymbol{\mu})^T \mathbf{W}(\boldsymbol{\theta} - \boldsymbol{\mu})$;
Obtenha a decomposição espectral de $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$;
Aplique $\mathbf{U}$ nos valores observados de $\boldsymbol{\theta}$, $\boldsymbol{\theta}^{**} = \mathbf{U}\boldsymbol{\theta}$;
**Saída:**
O conjunto de dados rotacionados $\boldsymbol{\theta}^{**}$.

## 2.5 Redes Neurais Artificiais

Uma grande vantagem de usar uma rede neural é a capacidade de resolver problemas sem a necessidade de definição de modelos explícitos. Isto possibilita detectar implicitamente qualquer relação não-linear entre a variável resposta e as variáveis explicativas. Além disso, não há necessidade de independência e normalidade das variáveis em estudo. Por isso, esta metodologia será utilizada para estimação das condicionais completas em nosso estudo.

Com o objetivo de simular a capacidade do cérebro humano de adquirir conhecimento, por meio da aprendizagem, o desenvolvimento das redes neurais artificiais (RNAs) foi inspirado na estrutura e funcionamento do sistema nervoso, que possibilita aos humanos lidar com informações imprecisas e de fazer generalizações para essas informações (Faceli et al., 2011).

Para Haykin (2009) as redes neurais assemelham-se com o cérebro em dois aspectos: o conhecimento é adquirido pela rede através de um processo de aprendizado e os conhecimentos adquiridos são armazenados nos pesos das conexões interneurônios, conhecidos como pesos

sinápticos. Dada suas propriedades, torna-se interessante a aplicabilidade de redes neurais em estudos de classificação de padrões e de previsão.

### 2.5.1 Sistema Nervoso Biológico

Entender o funcionamento do sistema nervoso biológico é de fundamental importância para o entendimento das redes neurais, pois elas tentam simular o comportamento do mesmo, conforme discutido em Silva, Spatti e Flauzino (2010).

O sistema nervoso é formado por um conjunto de células bastante complexo. Dentre elas os neurônios são as células mais conhecidas. Sendo consideradas como as unidades básicas de processamento do cérebro. Apresentam um papel fundamental no funcionamento e comportamento do corpo humano e do raciocínio.

As células neuronais controlam os sinais e ações dos seres vivos, ou seja, são responsáveis pela condução do impulso nervoso. Os neurônios, especializados na transmissão de informações, são compostos por três partes: dendritos, axônio e corpo celular ou soma.

### 2.5.2 Neurônio Artificial

O neurônio artificial é um modelo simplificado e simulado do neurônio biológico e seus atributos básicos são a adaptação e a representação de conhecimentos por meio conexões. Eles recebem sinais das variáveis de entrada e passam adiante uma versão ponderada e tratada desse sinal. Fazendo um paralelo com o sistema biológico, o contato dos dendritos com outros neurônios é representado por conexões ou sinapses, tendo como função tornar o sinal de saída de um neurônio em um sinal de entrada de outro. O modelo de neurônio mais simples e ainda um dos mais utilizado nas diferentes arquiteturas de RNA's foi proposto por McCulloch e Pitts (1943) é constituído basicamente de 7 partes conforme apresentado a seguir.

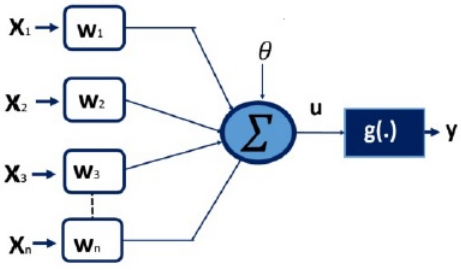



Figura 3: Neurônio Artificial.

(a) *Sinais de entrada* $\{x_1, \cdots, x_n\}$: valores assumidos pelas variáveis explicativas de uma aplicação;

(b) *Pesos Sinápticos* $\{w_1, \cdots, w_n\}$: têm por função ponderar cada uma das variáveis de entrada da rede, permitindo quantificar as suas respectivas relevâncias em relação à funcionalidade do respectivo neurônio;

(c) *Combinação Linear* $\mathbf{\Sigma}$: agrega todos os sinais de entrada que foram ponderados pelos respectivos pesos sinápticos produzindo um valor potencial de ativação;

(d) *Limiar de Ativação* $\boldsymbol{\theta}$: especifica qual o patamar apropriado para que o resultado da combinação linear ative a sinapse do neurônio;

(e) *Potencial de Ativação* $\mathbf{u}$: diferença entre o valor produzido pela combinação linear e o limiar de ativação; se $\mathbf{u} \leq \boldsymbol{\theta}$ diz-se que o neurônio apresentou potencial excitatório, caso contrário, o potencial apresentado será inibitório;

(f) *Função de Ativação* $\mathbf{g}()$: tem por objetivo limitar a saída do neurônio dentro de um intervalo de valores a serem assumidos pela própria imagem funcional;

(g) *Sinal de Saída* $\mathbf{y}$: valor final produzido pelo neurônio em relação a um determinado conjunto de sinais de entrada, podendo também ser usado por outros neurônios que estão interligados.

Em síntese, a expressão abaixo resume o resultado produzido pelo neurônio artificial proposto por McCulloch e Pitts (1943).

$$y = g(\sum_{i=1}^{n} w_i x_i - \theta). \tag{3}$$

Desta forma, pode-se resumir o funcionamento de um neurônio artificial da seguinte maneira: primeiramente, é apresentando um conjunto de valores de entrada ao mesmo, com esse conjunto de entrada, é realizada a multiplicação de cada entrada do neurônio pelo seu respectivo peso sináptico. Após as multiplicações, é então obtido o potencial de ativação do neurônio por meio da soma ponderada dos sinais de entrada subtraindo-se o limiar de ativação. Por fim, é aplicada a função de ativação neural ao seu potencial de ativação para obtenção do sinal de saída.

A partir da estrutura e funcionamento do neurônio biológico, diversos tipos de redes neurais que buscam simular este sistema em computador, têm sido apresentado na literatura como alternativas às técnicas de predição tradicionais.

Uma rede neural é uma estrutura de processamento de informação, distribuída paralelamente. Portanto, pode ser representada por uma estrutura genérica através de um grafo direcionado. No grafo os neurônios correspondem aos nós, as arestas funcionam como as sinapses. É possível representar também as camadas, que, juntamente com os neurônios, definem a arquitetura de uma RNA que é relacionada ao tipo, número de unidades de processamento (camadas), forma de conexão dos neurônios e tipo de aprendizagem em que são definidas as regras para o ajuste dos pesos da rede.

### 2.5.3 Camadas de uma Rede Neural

Uma RNA apresenta três tipos de camadas que podem ser definidas como:

- Camada de Entrada: recebe os dados/padrões a serem analisados e os distribui (sem modificá-los) a todos os neurônios da camada seguinte;
- Camadas Intermediárias, Escondidas ou Ocultas: responsáveis pela maior parte do processamento dos dados, através das conexões ponderadas, possuem a finalidade de extrair as informações/ características dos dados;
- Camada de Saída: recebe os estímulos das camadas anteriores e ativa uma resposta adequada.

### 2.5.4 Aprendizado de uma Rede Neural

O aprendizado de uma rede é feito por meio de um processo iterativo de ajustes aplicado a seus pesos até que se atinja uma solução generalizada para uma classe de problemas (Haykin, 2009). Os tipos de aprendizagem são classificados como:

- Aprendizado Supervisionado: deve-se dispor do conjunto de dados e das respectivas saídas, ou seja, é necessário indicar à rede a resposta desejada para o padrão de entrada;
- Aprendizado Não Supervisionado: a saída deve se auto organizar em relação às especificidades do conjunto de dados e assim identificar subconjuntos similares. Não há indicação da resposta desejada para os padrões de entrada;

- Reforço: o sistema aprende por meio de tentativa e erro, no qual o algoritmo recebe um feedback da análise de dados.

Há redes que conseguem classificar apenas objetos que são linearmente separáveis como as do tipo perceptron e adaline que são formadas por apenas uma camada oculta.

Já as redes do tipo perceptron multicamadas podem ser utilizadas para solução de problemas não linearmente separáveis (Faceli et al., 2011).

A rede neural mais simples, de uma camada, é chamada de Perceptron e é inspirada nos neurônios. É constituída de nós de entrada, os pesos de cada entrada, função de ativação e os nós de saída.

Funciona basicamente da seguinte maneira: é feita uma soma ponderada dos dados de entrada (que sempre são numéricos) com seus pesos, o resultado dessa soma passa pela função de ativação que vai gerar o resultado na saída.

A rede neural multicamadas é composta de camadas alinhadas de neurônios, na qual a primeira camada apresenta as informações de entrada (variáveis explicativas), que são distribuídas para as camadas intermediárias, denominadas ocultas. E por fim, temos a camada de saída (variável resposta), responsável por armazenar o resultado da rede. A Figura 4 ilustra o que foi explanado.

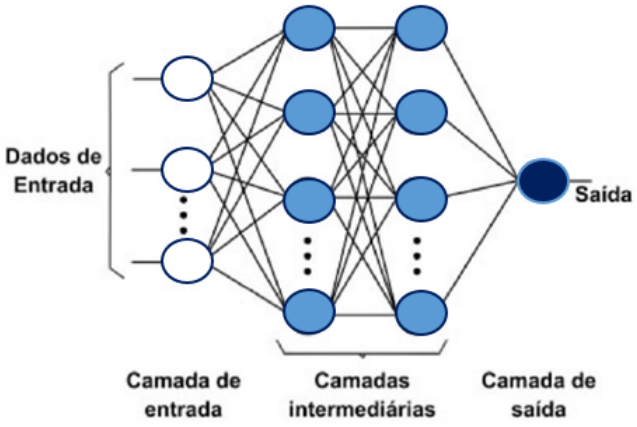



Figura 4: Exemplo de uma rede neural multicamadas.

A rede neural multicamadas é treinada de forma supervisionada, por meio do algoritmo backpropagation, conhecido também como regra Delta generalizada, em dois passos. Primeiro, um padrão é apresentado à camada de entrada da rede, então as informações/características dos dados são extraídas, camada por camada, até que a resposta seja produzida pela camada de saída. No segundo passo, a saída obtida é comparada ao resultado esperado para esse padrão

particular. O erro é calculado, e se for maior que zero é propagado a partir da camada de saída até a camada de entrada, e os pesos das conexões das unidades das camadas ocultas são ajustados de acordo com uma regra de correção de erro. Ou seja, são utilizados para ajustar os pesos e limiares de todos os seus neurônios.

### 2.5.5 Arquitetura de uma Rede Neural

Define a forma como seus neurônios estão organizados, uns em relação aos outros. Tal organização dár-se pela forma do direcionamento das conexões sinápticas dos neurônios. A literatura aponta três tipos de arquitetura como principais: a arquitetura feedfoward, em que o fluxo de informações é unidirecional, ou seja, da camada de entrada para camada de saída; a arquitetura recorrente, sua principal característica reside no fato que as saídas dos neurônios serem realimentadas como sinais de entrada para outros neurônios. Por fim, arquitetura em estrutura reticulada, Consideram a disposição espacial dos neurônios para extração de características, ou seja, sua localização espacial serve para ajuste de seus pesos e limiares.

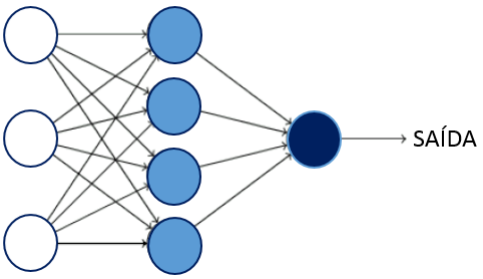



Figura 5: Exemplo de uma rede feedfoward.

Dentre as arquiteturas apresentadas, a *feedfoward* é a mais empregada em problemas de regressão e ela será a adotada neste trabalho. Para obter uma discussão detalhada sobre as arquiteturas apresentadas ver Silva, Spatti e Flauzino (2010).

Segundo Silva, Spatti e Flauzino (2010):

> *"As aplicações sucessivas das fases forward e backward fazem com que os pesos sinápticos e limiares dos neurônios se ajustem automaticamente em cada iteração, implicando-se na gradativa diminuição da soma dos erros produzidos pelas respostas da rede frente àquelas desejadas."*

### 2.5.6 Redes Multicamadas

As redes multicamadas (redes *multilayers perceptron*) apresentam arquitetura *feedfoward* e são caracterizadas pela presença de pelo menos uma camada intermediária, a Figura 5 é um exemplo. Apresentam aplicações nas mais variadas áreas do conhecimento como, por exemplo, otimização de sistemas, reconhecimento de padrões, aproximação universal de funções, cuja é a finalidade do seu uso neste trabalho, dentre outras.

Conforme já discutido na seção anterior, pode-se perceber que as redes multicamadas são constituídas por três partes: a *camada de entrada*, recebe os dados/padrões a serem analisados e os distribui (sem modificá-los) a todos os neurônios da camada seguinte; *camadas intermediárias*, responsável pela maior parte do processamento dos dados, através das conexões ponderadas, possui a finalidade de extrair as informações/ características dos dados; e a *camada de saída*, responsável por receber os estímulos das camadas anteriores e ativar uma resposta adequada. Nota-se ainda que, com exceção da camada de entrada, todas as outras camadas estão constituídas por neurônios, as quais implicam em um esforço computacional.

Cabe ressaltar que a configuração topológica da rede, tais como a quantidade de camadas intermediárias e seus respectivos números de neurônios depende de diversos fatores. Por exemplo, a classe de problema a ser tratada, a disposição espacial das amostras de treinamento e os valores iniciais atribuídos tanto aos parâmetros de treinamento como às matrizes de pesos (Silva, Spatti e Flauzino, 2010).

Uma vez definida a topologia da rede, o processo de treinamento para arquitetura feedfoward é feito de forma supervisionada, ou seja, para cada amostra de entrada, obtêm-se a respectiva saída desejada. Esse processo pode ser realizado por meio da utilização do algoritmo de treinamento denominado *backpropagation*, discutido a seguir.

### 2.5.7 Processo de Treinamento da Rede

Responsável pelo aprendizado da rede, o backpropagation tem como objetivo otimizar os pesos para que a rede neural aprenda a mapear as entradas para as saídas de forma correta. Ou seja, os pesos determinam o impacto que o valor de cada variável de entrada tem no valor das variáveis de saída.

O processo começa com a inicialização dos pesos da rede com pequenos valores aleatórios, com base nos valores de entrada a previsão da rede neural em alguns dados é observada e o valor da função erro entre o valor obtido e esperado para saída é calculado. Na busca de minimizar o

valor da função de erro, calculam-se os valores dos gradientes para cada peso da rede, e a partir do vetor gradiente calculado, os peso são atualizados de modo iterativo, até que o erro atinja um valor abaixo de algum limiar definido, ou atinja o número máximo de iterações preestabelecido.

De acordo com Deisenroth, Faisal e Ong (2020) o valor da função $\mathbf{y}$ em uma rede neural pode ser expresso por:

$$\mathbf{y} = (g_k \circ g_{k-1} \circ \cdots \circ g_1)(\mathbf{x}) = g_k(g_{k-1}(\cdots(g_1(\mathbf{x}))\cdots)),$$

ou seja, uma composição de funções, em que, $\mathbf{x}$ são as entradas, $\mathbf{y}$ são os valores observados e cada função $g_i,\ i = 1, \cdots, k$, possui seus próprios parâmetros.

Redes multicamadas apresentam funções $g_i(\mathbf{x}_{i-1}) = \sigma(\mathbf{W}_{i-1}\mathbf{x}_{i-1} + \mathbf{b}_{i-1})$ na $i$-ésima camada. $\mathbf{x}_{i-1}$ é a saída da camada $(i-1)$-ésima e $\sigma$ é uma função de ativação. Para treinar esses modelos, exige-se a obtenção do gradiente de uma função de perda $L$ em relação a todos os parâmetros do modelo $\mathbf{W}_j,\ \mathbf{b}_j,\ j = 1, \ldots, k$. Desta forma, é necessário computar o gradiente de $L$ em relação às entradas de cada camada. Por exemplo, considerando as entradas $\mathbf{x}$ e observações $\mathbf{y}$, uma estrutura de rede definida por

$$\begin{aligned}
&\mathbf{g}_0 := \mathbf{x} \\
&\mathbf{g}_i := \sigma_i(\mathbf{W}_{i-1}\mathbf{g}_{i-1} + \mathbf{b}_j),\ \ i = 0, 1, \ldots, k.
\end{aligned}$$

O interesse é encontrar $\mathbf{W}_j, \mathbf{b}_j,\ j = 0, 1, \ldots, k-1$, de modo que a perda quadrada

$$L(\boldsymbol{\theta}) = ||\mathbf{y} - \boldsymbol{g}_K(\boldsymbol{\theta}, \mathbf{x})||^2$$

seja minimizada, $\boldsymbol{\theta} = \{\mathbf{W}_0, \mathbf{b}_0, \ldots, \mathbf{W}_{k-1}, \mathbf{b}_{k-1}\}$.

Para obtenção dos gradientes em relação ao conjunto de parâmetros $\boldsymbol{\theta}$, é necessário obter as derivadas parciais de $L$ em relação aos parâmetros $\boldsymbol{\theta}_j = \mathbf{W}_j, \mathbf{b}_j$ de cada camada $j = 0, \ldots, k-1$. Fazendo uso da regra da cadeia para derivação, é possível determinar as derivadas parciais como

$$\frac{\partial L}{\partial \boldsymbol{\theta_i}} = \frac{\partial L}{\partial \mathbf{g}_k}\frac{\partial \mathbf{g}_k}{\partial \mathbf{g}_{k-1}} \cdots \frac{\partial \mathbf{g}_{i+2}}{\partial \mathbf{g}_{i+1}}\frac{\partial \mathbf{g}_{i+1}}{\partial \boldsymbol{\theta}_i}$$

Supondo que já se tenham obtido as derivadas parciais $\partial L/\partial \boldsymbol{\theta}_{i+1}$, então a maior parte do cálculo pode ser reutilizado para calcular $\partial L/\partial \boldsymbol{\theta}_i$. Neste ponto percebe-se o porquê do nome do algoritmo.

Uma vez obtido o gradiente da função custo, o ajuste dos valores do conjunto de parâmetros $\boldsymbol{\theta}$ é feito na direção oposta do mesmo. Deisenroth, Faisal e Ong (2020) apresentam uma discussão mais detalhada sobre o algoritmo.

### 2.5.8 Hiperparâmetros de uma Rede Neural

Para o processamento de uma rede neural é necessário a definição dos seguintes hiperparâmetros:

- Número de camadas: deve-se optar pelo menor número de camadas escondidas para solucionar o problema, o que pode ser avaliado a partir das métricas de desempenho aplicadas aos dados de validação;

- Número de neurônios: deve estar entre o tamanho da camada de entrada e o da camada de saída.

- Função de Ativação $\mathrm{g}()$ tem por objetivo limitar a saída do neurônio dentro de um intervalo de valores razoáveis a serem assumidos pela própria imagem funcional. Como exemplo, temos as funções *relu ("Rectified Linear Unit"), tanh (tangente hiperbólica) e sigmoid* ;

- Percentual dos dados alocados para os conjuntos de treinamento e teste.

- Tamanho do lote (*Batch Size*) - define o número de amostras que serão propagadas pela rede. O Batch Size pode ser uma das três opções:

  ***batch mode***: o tamanho do lote é igual ao conjunto de dados total. Tamanhos maiores de lotes levam a uma precisão menor nos dados de teste;

  ***mini-batch mode***: o tamanho do lote é maior que um, mas menor que o tamanho total do conjunto de dados;

  ***stochastic mode***: o tamanho do lote é igual a um. Portanto, o gradiente e os parâmetros da rede neural são atualizados após cada amostra;

- Número de épocas de treinamento - quantas passagens completas do conjunto de dados (épocas) devem ser realizadas;

- A compilação serve para configurar o processo de aprendizagem, nesta etapa é preciso definir três itens:

***Função Loss***: mede a diferença entre os dados de teste e os dados de validação. O objetivo é minimizar a função para guiar o modelo para direção certa;

***Optimizer***: define como os pesos da rede neural são atualizados com base no dados e sua função loss;

***Métrics***: usadas para avaliação, monitora os passos de treinamento e teste da rede.

### 2.5.9 Generalizando uma Rede Neural Simples

Um fato bastante curioso sobre a redes neurais é o fato de algumas configurações se assimilarem a alguns modelos matemáticos específicos. Por exemplo, ao adotar-se a função de ativação sinal no neurônio de McCulloch e Pitts, apresentado em (3), a seguinte regra de decisão é obtida

$$y = \begin{cases} 1, \sum_{i=1}^{n} w_i x_i \geq \theta \\ -1, \sum_{i=1}^{n} w_i x_i < \theta \end{cases} ,$$

que, conforme pode ser observado, se trata de um discriminador linear, utilizado para resolver problemas de classificação.

Um segundo exemplo a ser citado é a utilização da função de ativação identidade ao mesmo neurônio, para obtenção de um modelo de regressão linear

$$y_i = \sum_{j=1}^{n} w_j x_j + \epsilon_i \, .$$

Por fim, modelos lineares generalizados (mlg) são obtidos de maneira semelhante.

Retornando ao exemplo em que o neurônio, também considerado uma rede - a perceptron simples, assume a forma de um discriminador linear, fica meio óbvio que em regiões não linearmente separáveis a classificação dada não será a ideal. Isso é bastante discutido no problema de classificação para disjunção exclusiva (XOR). Tal disjunção é uma operação lógica entre dois operandos, apresentando valor lógico verdadeiro somente quando os dois operandos forem diferentes. Ao fazer a representação gráfica desse problema fica mais claro o que está sendo elucidado. A Figura 6 busca apresentar um possível resultado da rede para o operador, além disso, ela também apresenta um resultado para o operador AND e para o operador OR, os quais são linearmente separáveis. Foi assumido 1 para o valor lógico VERDADEIRO e 0 para FALSO.

Conforme pode ser observado na Figura 6, e discutido no parágrafo anterior, o perceptron

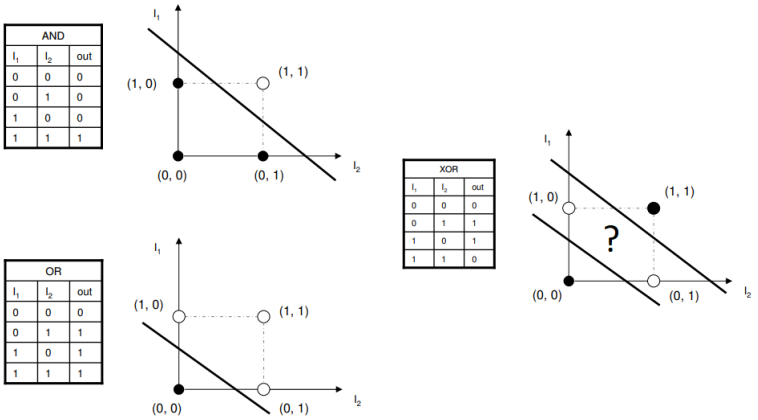



Figura 6: Perceptron como discriminador linear

simples não realizará a classificação de maneira ideal, além disso, o algoritmo de estimação dos pesos não convergirá, mais detalhes são apresentados em Silva, Spatti e Flauzino (2010).

Uma possível resolução para o problema XOR é a adição de uma camada oculta de dois neurônios a rede perceptron, a qual passará a ter a estrutura apresentada na Figura 7.

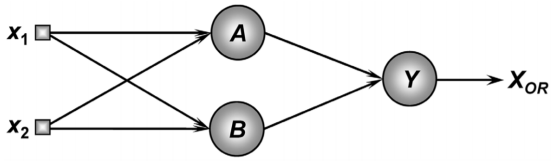



Figura 7: Perceptron Multicamada para o problema XOR

Matematicamente a rede pode ser expressa por

$$y = \sum_{j=1}^{2} v_j g_j \left( \sum_{i=1}^{2} w_{ji} x_i \right),$$

em que as função de ativação $g_i's$ são funções sigmoidais $f(x) = (1 + e^x)^{-1}$, responsáveis pelo caráter não linear da rede. Com isso, uma possível solução para o problema XOR é apresentada na Figura 8.

Através da Figura 8 percebe-se que a adição da camada oculta realiza uma operação de conjunção booleana (AND) em que o neurônio A apresentará valor igual 1 para padrões acima de sua reta e o B para abaixo da sua. Desta forma, a resposta da rede será igual a 1 somente quando os neurônios A e B forem iguais a 1.

Silva, Spatti e Flauzino (2010) deduzem que uma rede perceptron de duas camadas, sendo uma delas a camada escondida e a outra a própria camada de saída, é capaz de mapear qualquer

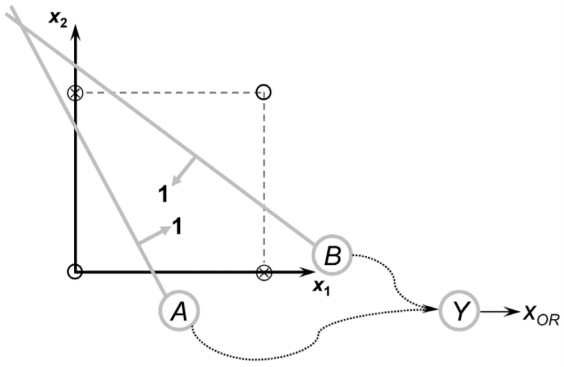



Figura 8: Resolução do problema XOR

problema de classificação de padrões cujos elementos estejam dentro de uma região convexa.

## 2.6 Regressão Quantílica

Na seção anterior foram apresentados alguns exemplos de redes neurais e mostrado que elas se assimilam a alguns modelos matemáticos, ou estatísticos, no caso o discriminador linear e os modelos de regressão. Como o presente trabalho busca empregar a combinação das redes neurais e da regressão quantílica, é feita uma revisão desta a seguir.

O uso de regressão quantílica se torna interessante, pois ao trabalhar com modelos flexíveis para os quantis, ao invés de para as médias e variâncias, conseguimos melhores aproximações para as DCC's, reduzindo, em última instância, o erro da estimativa da distribuição à posteriori de interesse.

Segundo Koenker (2005) a regressão ajustada pelo método dos mínimos quadrados ordinários é uma visão limitada de um conjunto de distribuições, pois se baseia na média, informação resumida e incompleta de uma distribuição. Uma alternativa mais completa ajustar diversas curvas de regressão para diferentes quantis associados.

O modelo de regressão quantílica, em vez de avaliar apenas o impacto dos **X** no **Y** médio como é feito numa regressão linear, verifica o efeito que os preditores **X** terão sobre os quantis de **Y**, neste caso são estimadas várias retas para diferentes quantis associados. Essa abordagem é interessante para variáveis dependentes cuja distribuição apresenta assimetria, caudas pesadas ou heteroscedasticidade, pois permite captar efeitos ao longo de toda a distribuição da variável dependente.

Segundo He (1997) as curvas de percentil são geralmente calculadas um nível de cada vez e um problema preocupante com a regressão quantílica é que as curvas de quantis podem se

cruzar, fenômeno do cruzamento de quantis. De certa forma, o cruzamento de curvas de quantis reflete uma escassez de dados na região do percentil de interesse. O problema de cruzamento pode ser evitado ao forçar ordenação das curvas.

Para o intuito desde trabalho, é suficiente definir o problema de regressão quantílica à luz de um contexto de otimização. Rasteiro (2017) tráz uma discussão completa sobre o tópico.

O modelo para regressão quantílica pode ser definido como

$$y_i = \mathbf{x}_i' \boldsymbol{\beta}_q.$$

Desta forma, para estimar o q-ésimo quantil, é necessário encontrar $\boldsymbol{\beta}_q$ de tal forma que a condição abaixo seja satisfeita

$$P(y \leq \mathbf{x}_i' \boldsymbol{\beta}_q | \mathbf{x}) = q,$$

Assim, a obtenção das estimativas $\boldsymbol{\beta}_q$ é realizada através da minimização da função definida em (4).

$$\mathcal{L}_n(\boldsymbol{\beta}_q | \mathbf{y}, \mathbf{X}) = \sum_{i:e_{i,q} \geq 0}^{n} q|y_i - \mathbf{x}_i' \boldsymbol{\beta}_q| + \sum_{i:e_{i,q} < 0}^{n} (1-q)|y_i - \mathbf{x}_i' \boldsymbol{\beta}_q|, \tag{4}$$

em que $e_{i,q} = y_i - \mathbf{x}_i' \boldsymbol{\beta}_q$, e $\boldsymbol{\beta}_q$ representa o efeito marginal das variáveis explicativas $\mathbf{X}$ no q-ésimo quantil de $\mathbf{Y}$.

### 2.6.1 Um Pouco Sobre a Função de Perda da Regressão Quantílica

Segundo Abeywardana (2018) a função de perda mais comumente usada, em problemas de regressão, é a função de erro quadrático médio e ao se exponencializar o negativo dessa função, o resultado será a distribuição gaussiana, sendo que a moda desta distribuição corresponde ao parâmetro média $\mu$. Na regressão quantílica, a perda para uma observação individual é definida como:

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

em que $q$ é o quatil requerido e $e_i$ é a diferença entre o valor observado e o predito pelo modelo, $e_i = y_i - f(\mathbf{x}_i)$. A perda média sob os dados de entrada é dada por

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

Ao se tomar o negativo da perda individual e exponenciá-la, o que se obtém é a distribuição de Laplace assimétrica. Conforme pode ser observado na Figura 9, o motivo pelo qual essa função de perda funciona é que, ao buscar a área sob o gráfico da função à esquerda de zero, ela corresponde a $q$, ou seja, o quantil necessário.

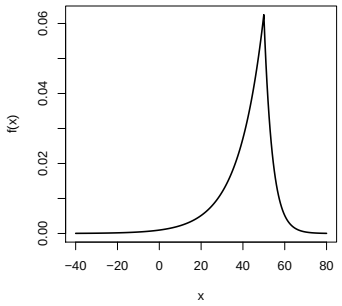



Figura 9: Função densidade de probabilidade de uma distribuição Laplace assimétrica.

No caso no qual $q$ é igual a 0,5 a função perda é equivalente ao Erro Médio Absoluto (MAE) e estima a mediana (50º percentil), em vez da média (Abeywardana, 2018).

## 2.7 Regressão Quantílica via Redes Neurais

A regressão quantílica via redes neurais é uma extensão não linear da regressão quantílica. Pode ser estruturada por uma camada de entrada (os preditores), uma ou mais camadas ocultas e uma camada de saída (a previsão para um quantil). Nas camadas ocultas, ajustamos a entrada em direção à saída pela multiplicação com pesos, adicionando algum viés(neurônio de polarização) e aplicando uma função de ativação não linear.

Sem perda de generalidade, substituindo-se $\boldsymbol{\beta}_q$ por $\mathbf{W}_q$ em (4), Taylor (2000), utiliza uma rede feedfoward para estimativa de quantis de modelos não lineares. No texto, o autor adota um modelo de rede neural com uma camada oculta para ajuste de uma regressão quantílica

$$f(\mathbf{x}_t, \mathbf{v}, \mathbf{w}) = g_2\left(\sum_{j=0}^{m} v_j g_1\left(\sum_{i=0}^{n} w_{ji} x_{it}\right)\right).$$

Em que, $g_1(.)$ e $g_2(.)$ são funções de ativação e $w_{ji}$ e $v_j$ são os pesos(parâmetros) a serem estimados. O modelo com $n$ camadas ocultas, em notação matricial, é descrito abaixo

$$f(\mathbf{X}_t, \mathbf{W}_1, \cdots, \mathbf{W}_n) = g_n\left(\mathbf{W}_n g_{n-1}\left(\mathbf{W}_{n-1} g_{n-2}\left(\cdots g_2 \mathbf{W}_2\left(g_1(\mathbf{W}_1 \mathbf{X}_t)\right)\cdots\right)\right)\right).$$

Tão logo, fazendo-se as devidas substituições, o q-ésimo quantil é obtido minimizando (4)

$$\min_{\mathbf{W}_1,\ldots,\mathbf{W}_n} \mathcal{L} = \Bigg( \sum_{t \mid y_t \leq f(\mathbf{X}_t,\mathbf{W}_1,\ldots,\mathbf{W}_n)} q \mid y_t - f(\mathbf{X}_t, \mathbf{W}_1, \ldots, \mathbf{W}_n) \mid + \\ + \sum_{t \mid y_t < f(\mathbf{X}_t,\mathbf{W}_1,\ldots,\mathbf{W}_n)} (1-q) \mid y_t - f(\mathbf{X}_t, \mathbf{W}_1, \ldots, \mathbf{W}_n) \mid \Bigg).$$

A escolha do número de camadas e do número de neurônios em cada camada pode ser feita via validação cruzada e inspeção do erro quadrático médio.

## 2.8 Interpolação Monotônica

Conforme abordado na seção acima, por meio da regressão quantílica pretende-se estimar alguns pontos da função de distribuição, $\hat{F}(x_k), k = 1, 2, \cdots, n$. Porém, mesmo para um $n$ grande, não se garante a estimação da curva. Desta forma, faz-se necessário o uso de algum método que permita uma boa aproximação para $F$ com os pontos obtidos.

### 2.8.1 *Splines*

Uma possível técnica que permite uma aproximação para $F$ é fazer uso de uma combinação linear de funções base,

$$F(x) \approx f_k(x) = \sum_{i=1}^{k} \theta_j \beta_j(x),$$

em que $\beta_1, \cdots, \beta_k$ são funções base conhecidas e $\theta_1, \cdots, \theta_k$ são valores dos coeficientes a serem estimados.

Uma família de funções base bastante empregada na literatura é a família de funções *splines*. A ideia por trás da construção deste método é particionar o intervalo da função a ser estimada, digamos $[a, b]$, em $k$ subintervalos, $[\epsilon_{i-1}, \epsilon_i], 1 \leq i \leq k$, tal que $a \leq \epsilon_0 \leq \cdots \leq \epsilon_k \leq b$. Uma vez realizado tal particionamento, para cada subintervalo, ajusta-se um polinômio $p_i$ como aproximação de $F$ para o subintervalo em questão. Os valores $\epsilon_0, \cdots, \epsilon_k$ são conhecidos como nós (*knots*).

O problema ao adotar este procedimento, é que seu resultado é uma função de aproximação polinomial por partes, $s(\cdot)$, e como os polinômios $p_i$ são construídos de forma independentemente, ela não é uma função contínua. Desta forma, é necessário que as partes dos polinômios sejam unidas de forma suave nos nós $\epsilon_1, \cdots, \epsilon_{k-1}$, além disso, é necessário que sejam deri-

váveis um certo número de vezes. Obtêm-se então uma função polinomial por partes, suave, conhecida por função *spline*.

Um *spline* de ordem $m$ com $k-1$ nós internos em $\epsilon_1, \cdots, \epsilon_{k-1}$ é qualquer função da forma

$$s(x) = \sum_{i=0}^{m-1} c_i x_i + \sum_{j=1}^{k-1} d_j (x - \epsilon_j)^{m-1} I(x \geq \epsilon_j)$$

em que $c_1, \cdots, c_{m-1}$ e $d_1, \cdots, d_{k-1}$ são números reais.

Fritsch e Carlson (1978), apresentam um método para interpolação unidimensional usando interpolantes cúbicos por partes, uma classe especial de *splines*. Desta forma, sejam $(x_i, y_i), i = 1, \cdots, n$ os pares aos quais se deseja realizar a interpolação. A aplicação do método retorna como valor interpolado o valor $y'$ dado por

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

em que $h_i = x_{i+1} - x_i$ e $d_i = \frac{y_{i+1} - y_i}{h_i}$.

Este método consiste basicamente em obter uma média harmônica ponderada das inclinações. Mais detalhes sobre a técnica *splines* pode ser obtido em Morellato (2014).

No *splines* monotônico a inclinação em cada ponto do *grid* é determinada de forma a garantir um comportamento monotônico da função de interpolação.

# 3 Amostrador de Gibbs Aproximado sem Verossimilhança

Nos casos em que as distribuições condicionais exatas não podem ser computadas de forma analítica, uma opção é modelar empiricamente as condicionais por meio de modelos de regressão e posteriormente utilizá-las em um amostrador Gibbs, evitando assim a necessidade de se amostrar diretamente da distribuição exata (Rodrigues, Nott e Sisson, 2019).

## 3.1 Método proposto por Rodrigues, Nott e Sisson (2019)

Para construir o amostrador de Gibbs aproximado sem verossimilhança, o algoritmo começa de maneira semelhante a muitos algoritmos ABC, extraindo amostras $\{(\boldsymbol{\theta}^{(i)}, \boldsymbol{s}^{(i)})\}_{i=1}^{N}$ da distribuição $p(\boldsymbol{X}|\boldsymbol{\theta})b(\boldsymbol{\theta})$ e computando $\boldsymbol{s}^{(i)} = S(\boldsymbol{X}^{(i)})$, em que $b(\boldsymbol{\theta})$ é uma distribuição arbitrária - normalmente definida como a própria distribuição a priori $\pi(\boldsymbol{\theta})$.

Uma vez as amostras obtidas, são ajustados modelos de regressão da forma $\theta_d|(\boldsymbol{S}, \boldsymbol{\theta}_{-d}) \sim f(\theta_d|\boldsymbol{\beta}_d^{+1}, g_d(\boldsymbol{S}, \boldsymbol{\theta}_{-d}))$, em que $\boldsymbol{\theta}_{-d}$ é o vetor $\boldsymbol{\theta}$ excluindo $\theta_d$. Desta maneira, garante-se que $f(\cdot)$ seja o mais próximo possível de $\pi(\theta_d|\boldsymbol{s}_{\text{obs}}, \boldsymbol{\theta}_{-d})$, a verdadeira distribuição condicional completa de $\boldsymbol{\theta}_d$. As funções $g_d(\boldsymbol{S}, \boldsymbol{\theta}_{-d})$ indicam funções de $\mathbf{S}$ e $\boldsymbol{\theta}_{-d}$ usadas no modelo de regressão para determinar a distribuição condicional de $\theta_d$. Esses modelos são ajustados usando as amostras ponderadas $\{(\boldsymbol{\theta}^{(i)}, \boldsymbol{s}^{(i)}, w_d^{(i)})\}_{i=1}^N$, em que os pesos $w_d^{(i)} \propto K_h(||\boldsymbol{s}^{(i)} - \boldsymbol{s}_{\text{obs}}||)\pi(\boldsymbol{\theta})/b(\boldsymbol{\theta})$ garantem que seja dada maior importância às amostras mais próximas aos dados observados $\boldsymbol{s}_{\text{obs}}$.

Em seguida, um procedimento padrão do amostrador de Gibbs aproximado é implementado amostrando cada parâmetro, por sua vez, de uma aproximação à sua respectiva distribuição condicional completa $\theta_d^{(m)}|(\boldsymbol{s}_{\text{obs}}, \boldsymbol{\theta}_{-d}) \sim f(\theta_d|\hat{\boldsymbol{\beta}}_d^{+}, g_d(\boldsymbol{s}_{\text{obs}}, \boldsymbol{\theta}_{-d}))$ para $d = 1, \ldots, D$. Se $f(\theta_d|\hat{\boldsymbol{\beta}}_d^{+}, \boldsymbol{s}_{\text{obs}}, \boldsymbol{\theta}_{-d}) = \pi(\theta_d|\boldsymbol{s}_{\text{obs}}, \boldsymbol{\theta}_{-d})$ para todo $d$, o amostrador Gibbs resultante terá como alvo exatamente $\pi(\boldsymbol{\theta}|\boldsymbol{s}_{\text{obs}})$. A implementação do método original é realizada no Algoritmo 6.

**Algoritmo 6:** Amostrador de Gibbs aproximado, sem verossimilhança - Método Original

**Entrada:**

Um conjunto de dados observados ($\boldsymbol{X}_{\text{obs}}$);
Uma priori $\pi(\theta)$ e um modelo generativo intratável $p(\boldsymbol{X}|\theta)$;
Uma distribuição $b(\boldsymbol{\theta})$ descrevendo uma região de alta densidade a posteriori;
Um vetor observado de medidas resumo $\boldsymbol{s}_{\text{obs}} = \boldsymbol{S}(\boldsymbol{X}_{\text{obs}})$;
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 $f(\theta_d|\boldsymbol{\beta}_d^+, g_d(\boldsymbol{S}, \boldsymbol{\theta}_{-d}))$ para aproximar cada distribuição condicional completa $\pi(\theta_d|\boldsymbol{s}_{\text{obs}}, \boldsymbol{\theta}_{-d})$ para $d = 1, \ldots, D$.

*Simulação de Dados Sintéticos*

**para** $i = 1, 2, \cdots, N$ **faça**

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

**fim**

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

*Estimação dos Modelos*

**para** $d = 1, 2, \cdots, D$ **faça**

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

**fim**

*Aproximação de Gibbs*

**para** $m = 1, 2, \cdots, M$ **faça**

**para** $d = 1, 2, \cdots, D$ **faça**

4.1 $\boldsymbol{\theta}_{-d}^\star = (\tilde{\theta}_1^{(m)}, \ldots, \tilde{\theta}_{d-1}^{(m)}, \tilde{\theta}_{d+1}^{(m-1)}, \ldots, \tilde{\theta}_D^{(m-1)})^\top$ o vetor que contém os valores atualizados de $\tilde{\theta}_j^{(\cdot)}$, $j \neq d$;
4.2 Atualização de Gibbs: Amostre $\tilde{\theta}_d^{(m)}|(\boldsymbol{s}_{\text{obs}}, \boldsymbol{\theta}_{-d}^\star) \sim f(\theta_d|\hat{\boldsymbol{\beta}}_d^+, g_d(\boldsymbol{s}_{\text{obs}}, \boldsymbol{\theta}_{-d}^\star))$;

**fim**

**fim**

**Saída:**

O vetor contendo amostras aproximadas de Gibbs para os parâmetros.

## 3.2 Método Proposto

Uma das desvantagens do método original, apresentado pelos autores, é a mistura possivelmente lenta na cadeia de Markov, em especial para modelos com fortes correlações à posteriori. Essa limitação é herdada naturalmente do próprio amostrador de Gibbs. Além disso, ao modelar apenas a média e a variância, as aproximações adotadas assumem implicitamente que a forma das distribuições condicionais completas é fixa em uma vizinhança de $s_{\text{obs}}$. Neste estudo, propomos (Algoritmo 7) uma abordagem para estimar densidade a posteriori, introduzindo duas inovações ao método apresentado por Rodrigues, Nott e Sisson (2019).

1. Na literatura encontra-se estudos em que as densidades condicionais completas são estimadas a partir do ajuste de modelos flexíveis de regressão não linear, como por exemplo Blum e Francois (2010) que utilizam redes neurais, para modelar tanto parâmetro de locação $\mu$ quanto de escala $\sigma$ . Entretanto, mesmo com essa abordagem mais flexível assume-se que a distribuição dos resíduos é a mesma em todo espaço das estatísticas resumo onde o peso é positivo, não conseguindo, portanto, acomodar variações de forma das distribuições condicionais completas.

   Dado o exposto, propõe-se então, aproximar as condicionais completas por meio do ajuste de regressão quantílica via redes neurais com correção *spline* monotônica, modelo que se destaca pela flexibilidade e capacidade de lidar com a complexidade e não-linearidades dos dados.

   Outras abordagens para estimar as densidades condicionais foram testadas inicialmente nesse estudo (resultados não apresentados) incluíndo *FlexCode* apresentado por Izbicki e Lee (2016) no qual, reformula a função densidade condicional como uma série ortogonal não paramétrica em que é realizada uma expansão ortonormal (como, por exemplo, a base de *Fourier*) dos coeficientes, que são estimados via regressão. Em nossas simulações, o método se apresentou lento e não apresentou ajustes suficientemente precisos. *A quantile regression forests* proposta por Meinshausen (2006) visa estimar os quantis condicionais dos dados, sendo um método baseado em florestas aleatórias. Novamente, não observamos bons resultados, mesmo tendo explorado diferentes configurações. *Monotone composite quantile regression neural network* (MCQRNN) proposta por Cannon (2018) é uma extensão não linear da regressão quantílica, composta por uma camada de entrada (os preditores), uma ou mais camadas ocultas e uma camada de saída (a previsão

para um vetor de quantis). O pacote *qrnn* do *software* R é limitado a 2 camadas ocultas, o que prejudica o ajuste do modelo conforme a dimensão e complexidade dos dados aumenta, não tendo apresentado, desta forma, resultado satisfatório em nossas simulações.

2. Descorrelação prévia dos parâmetros para acelerar a convergência e aumentar a velocidade de mistura (quão rápido uma cadeia consegue explorar completamente o suporte da distribuição alvo). Ou seja, ao realizar a construção dos modelos de regressão e do amostrador de Gibbs, no método proposto, pode-se optar por operar no espaço original ou em um espaço transformado, no caso da presença de forte autocorrelação entre os parâmetros. Kessy, Lewin e Strimmer (2018) apresentam diversos métodos que podem ser utilizados para descorrelação/*whitining* de variáveis.

   Para melhor entendimento do método proposto descrevemos a seguir os passos necessários detalhadamente.

### 3.2.1 Simulação de Dados

Semelhantemente ao método original, para construção do amostrador de Gibbs aproximado sem verossimilhança, o algoritmo começa extraindo amostras $\{(\boldsymbol{\theta}^{(i)}, \boldsymbol{s}^{(i)})\}_{i=1}^{N}$ da distribuição $p(\mathbf{X}|\boldsymbol{\theta})b(\boldsymbol{\theta})$ e computando $\boldsymbol{s}^{(i)} = \boldsymbol{S}(\mathbf{X}^{(i)})$, em que $b(\boldsymbol{\theta})$ é uma distribuição arbitrária.

### 3.2.2 Descorrelação dos Dados - Opcional

Para descorrelação dos dados há desde métodos simples, como os apresentados em Kessy, Lewin e Strimmer (2018), até complexos quanto um fluxo normalizador abordado por Kobyzev, Prince e Brubaker (2020).

Nesse estudo, procede-se com a descorrelação dos parâmetros $\boldsymbol{\theta}_d$ por meio da técnica PCA ponderada, com os pesos oriundos de uma estimativa preliminar (via ABC) da posteriori, gerando o novo vetor de parâmetros transformados $\check{\boldsymbol{\theta}}_d$, para os quais os modelos de regressão quantílica com correção *spline* monotônica via redes neurais são construídos.

### 3.2.3 Estimação dos Modelos Quantílicos

Uma vez definido se será ou não utilizada transformação dos parâmetros, adiciona-se a estrutura de uma rede neural ao modelo de regressão quantílica para estimar os quantis das distribuições condicionais completas $\boldsymbol{\tau}(\theta_d)|(\boldsymbol{S}, \boldsymbol{\theta}_{-d})$. Desta forma, $\boldsymbol{\tau}(\theta_d)$ provê a melhor aproximação

possível para $\pi(\boldsymbol{\theta}_d|\boldsymbol{s}_{\text{obs}}, \boldsymbol{\theta}_{-d})$. As variáveis dependentes variam com $d$, mas normalmente são de dimensões relativamente baixas (Rodrigues, Nott e Sisson, 2019). As funções $g_d(\boldsymbol{S}, \boldsymbol{\theta}_{-d})$ indicam funções de $\mathbf{S}$ e $\boldsymbol{\theta}_{-d}$ usadas na camada de entrada da rede neural, para determinar a distribuição condicional de $\theta_d$.

A implementação das redes neurais foi realizada no *software* R, por meio do pacote *Keras*, que utiliza como *backend* o pacote *Tensorflow*, desenvolvido pelo Google para implementação de redes neurais convencionais e profundas em diversas topologias. Para atingir os objetivos do trabalho foi necessário adaptar a função de perda da rede neural no pacote.

A Construção do modelo foi realizada de forma sequencial, por meio da função *"Keras-model-sequential()"*, ou seja, cada camada é adicionada uma após a outra, sendo que, o *output* da primeira camada serve como *input* da segunda.

É importante ressaltar que o pacote *Keras* apresenta a opção de se trabalhar com pesos para as amostras, no entanto, é limitada às funções de perda já inclusas no pacote. O que impossibilitou o seu uso no estudo, visto que, a função de perda utilizada foi customizada.

### 3.2.4 Aproximação Gibbs Sampling

Um procedimento padrão do amostrador de Gibbs aproximado é implementado amostrando cada parâmetro, por vez, de uma aproximação à sua respectiva distribuição condicional completa. Precisamente, por meio da regressão quantílica estima-se os pontos pertencentes ao grid definido para os quantis desejados da função de distribuição, $\hat{F}(x_k), k = 1, 2, \cdots, n$ em que $n$ é o número total de pontos a ser estimados.

Posteriormente, é realizada uma suavisação via *spline* monotônico para obtermos uma aproximação da DCC em todo o espaço (e não apenas sobre o *grid*). Esse procedimento garante que os quantis "não cruzam". Daí, utilizamos o método da transformada inversa para gerar uma amostra da DCC aproximada. É importante mencionar que o ajuste via *splines* foi realizado num espaço transformado, para garantir que a aproximação da distribuição $\mathrm{F}(\theta_d|\boldsymbol{s}_{\text{obs}}, \boldsymbol{\theta}_{-d})$ estivesse restrita ao intervalo 0 a 1.

### 3.2.5 Mudança de Escala

No caso em que se opte por descorrelacionar os parâmetros para aumentar a velocidade da mistura, é necessário realizar a transformação inversa de $\check{\boldsymbol{\theta}}$, para retorna à escala original $\boldsymbol{\theta}$.

**Algoritmo 7:** Amostrador de Gibbs aproximado, sem verossimilhança - Método Proposto

**Entrada:**

Um conjunto de dados observados ($\boldsymbol{X}_{\text{obs}}$);
Um vetor observado de medidas resumo $\boldsymbol{s}_{\text{obs}} = \boldsymbol{S}(\boldsymbol{X}_{\text{obs}})$;
*[Opcional]* Um kernel de suavização $K_h(u)$ com parâmetro de escala $h > 0$;
Uma função $\psi : [0,1]$ —-> $\mathbb{R}$, $x$ |—> $\psi$(x);
Um modelo generativo $\pi(\theta)p(\boldsymbol{X}|\theta)$;
Um inteiro positivo $N$ definindo o número de amostras sintéticas a serem geradas;
Um inteiro positivo $M$ definindo o número de iterações do amostrador Gibbs;
Um vetor $\boldsymbol{q} = (q_1, q_2, \ldots, q_k)$, em que, $0 < q_k < 1$, que define os quantis a serem estimados;

*Simulação de Dados Sintéticos*

**para** $i = 1, 2, \cdots, N$ **faça**

1.1 Gere $\boldsymbol{\theta}^{(i)} \sim b(\boldsymbol{\theta})$ de alguma distribuição adequada $b(\boldsymbol{\theta})$;
1.2 Gere $\boldsymbol{X}^{(i)} \sim p(\boldsymbol{X}|\boldsymbol{\theta}^{(i)})$ do modelo;
1.3 Calcular as medidas resumo $\boldsymbol{s}^{(i)} = \boldsymbol{S}(\boldsymbol{X}^{(i)})$;

**fim**

*[Opcional] Descorrelação via PCA Ponderada*

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

*Estimação dos Modelos Quantílicos*

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

*Aproximação Gibbs sampling*

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

**para** $m = 1, 2, \cdots, M$ **faça**

**para** $d = 1, 2, \cdots, D$ **faça**

4.2.1 Obtenha $\hat{\tau}_q(\theta_d)|\, g_d(\boldsymbol{s}_{\text{obs}}, \boldsymbol{\theta}^{\star}_{-d})$, em que $\boldsymbol{\theta}^{\star}_{-d} = (\tilde{\theta}_1^{(m)}, \ldots, \tilde{\theta}_{d-1}^{(m)}, \tilde{\theta}_{d+1}^{(m-1)}, \ldots, \tilde{\theta}_D^{(m-1)})^\top$ o vetor que contém os valores atualizados de $\tilde{\theta}_j^{(\cdot)}$, $j \neq d$;
4.2.2 Ajustar um spline 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 $\tilde{\theta}_d^{(m)} = \hat{F}^{-1}(\psi(u))$;

**fim**

**fim**

*[Opcional] Mudança de Escala*

4.3 Realizar transformação inversa de $\check{\boldsymbol{\theta}}$, para retorna à escala original $\theta$;

**Saída:**

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

# 4 Estudos de Simulação

Com intuito de avaliar o desempenho do método proposto em relação ao método apresentado por Rodrigues, Nott e Sisson (2019), foram realizados dois estudos de simulação: o primeiro estima a distribuição a posteriori de um modelo normal bivariado, no qual optou-se pelo processo em que há a etapa de descorrelação dos parâmetros; o segundo estima a distribuição a posteriori de um modelo de mistura gaussiana D-dimensional sem considerar a etapa de descorrelação. Em ambas as simulações utiliza-se regressão quantilica via redes neurais multicamadas, para estimação das condicionais completas.

## 4.1 Exemplo 1: Normal Bivariada com $\Sigma$ conhecida

Considere um modelo onde a variável aleatória bivariada, $\boldsymbol{X}$, segue uma distribuição normal bivariada com média $\mu = (\mu_1, \mu_2)$ e matriz de covariância, $\Sigma$, conhecida. Consideramos ainda uma priori conjugada para o vetor de médias, ou seja,

$$\mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$
$$\mu \sim U[-10, 10],$$

seja $x = (x_1, x_2)$ uma única amostra observada de $X$, temos

$$\pi(\boldsymbol{\mu}|\mathbf{x}) \sim N(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*),$$

em que, $\boldsymbol{\mu}^* = (\boldsymbol{\Sigma}_0^{-1} + \boldsymbol{\Sigma}^{-1})^{-1}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 + \boldsymbol{\Sigma}^{-1}\mathbf{x})$ e $\boldsymbol{\Sigma}^* = (\boldsymbol{\Sigma}_0^{-1} + \boldsymbol{\Sigma}^{-1})^{-1}$.

Neste exemplo, a verossimilhança pode ser facilmente avaliada, o que permite a comparação das distribuições estimadas com as respectivas distribuições exatas. Além disso, fazendo-se uso das propriedades da distribuição normal multivariada, a distribuição condicional completa para $\mu_1$ pode ser expressa por

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

A distribuição condicional completa para $\mu_2$ decorre de forma similar.

Os parâmetros utilizados para simulação foram, $\rho = 0,9$, $\sigma^2 = 1$ e $x = \boldsymbol{s}_{obs} = (5/2, 5/2)$, atribuiu-se uma distribuição uniforme como priori $U[-10, 10]$ para $\mu_1$ e $\mu_2$ .

Para obter $(\mu_1^i, \mu_2^i, s_1^i, s_2^i)$, $i = 1, ..., N$, foram geradas $N = 10.000$ amostras de todo o processo, considerando $b(\boldsymbol{\mu}) = \pi(\boldsymbol{\mu})$, e procedeu-se com uma rodada preliminar de ABC por importância com uma taxa de aceitação de 100%, visto que o objetivo era a estimação dos pesos para ajustar a PCA ponderada. Para isso, foi utilizado o pacote $abc$ do software R (Csillery, Francois e Blum, 2012). Em seguida, procedeu-se com a descorrelação dos parâmetros $\boldsymbol{\mu}_d$ por meio da técnica PCA ponderada (função $prcomp$ do *software* R), ajustada a partir do conjunto de amostras com seus respectivos pesos, gerando o vetor de parâmetros transformados $\boldsymbol{\mu}_d^*$.

Uma vez realizada a transformação nos parâmetros, modelos de regressão quantílica via redes neurais com correção *spline* monotônica foram construídos para os mesmos. Neste ponto há uma vantagem em relação ao método original, pois não é preciso definir manualmente a estrutura das covariáveis adotadas nos modelos de regressão usados para aproximar as DCC, o que segundo Rodrigues, Nott e Sisson (2019) é necessário para obter boas estimativas no método original. Nesta etapa, para modelar as distribuições condicionais de $\mu_1$, $\mu_2$, procedeu-se com a aproximação por meio do ajuste de regressões quantílicas (quantis no intervalo [0,002 ; 0,998]) via redes neurais com correção *spline* monotônica . Por se tratar de um exemplo meramente ilustrativo, optou-se por uma rede com apenas uma camada intermediária com 32 neurônios, função de ativação *relu*, *batch size* igual a 32 (valor padrão) e 50 épocas (definido a partir da análise do gráfico de perda da base de validação), para os demais hiperparâmetros foram considerados os valores *defaut* do pacote *Keras*.

Após ajustar modelos razoáveis para as distribuições condicionais, procede-se com a amostragem Gibbs ABC. Para implementação do algoritmo foram consideradas $M = 5.000$ iterações e utilizado como valor inicial $(\mu_1, \mu_2) = (0, 0)$. A Figura 6, apresenta os resultados obtidos a partir do método proposto, em que é possível verificar a consistência do método, visto que a estimativa para a posteriori obtida aproxima a verdadeira densidade, o que pode ser evidenciado na Figura 7(e) onde temos os pontos representando a distribuição a posteriori aproximada e os contornos representando a verdadeira distribuição a posteriori.

Para avaliar a efetividade do método proposto em relação ao método apresentado por Rodrigues, Nott e Sisson (2019) comparamos os resultados obtidos por cada um deles na Figura 11. Ao analisar os paineis (a) e (b), com as cinquenta primeiras iterações de Gibbs, percebe-se que a mistura na cadeia para o método original é mais lenta que para o método proposto. Esse comportamento também é observado na Figura 11c, em que as distâncias entre pontos consecutivos da cadeia é maior para o método proposto.

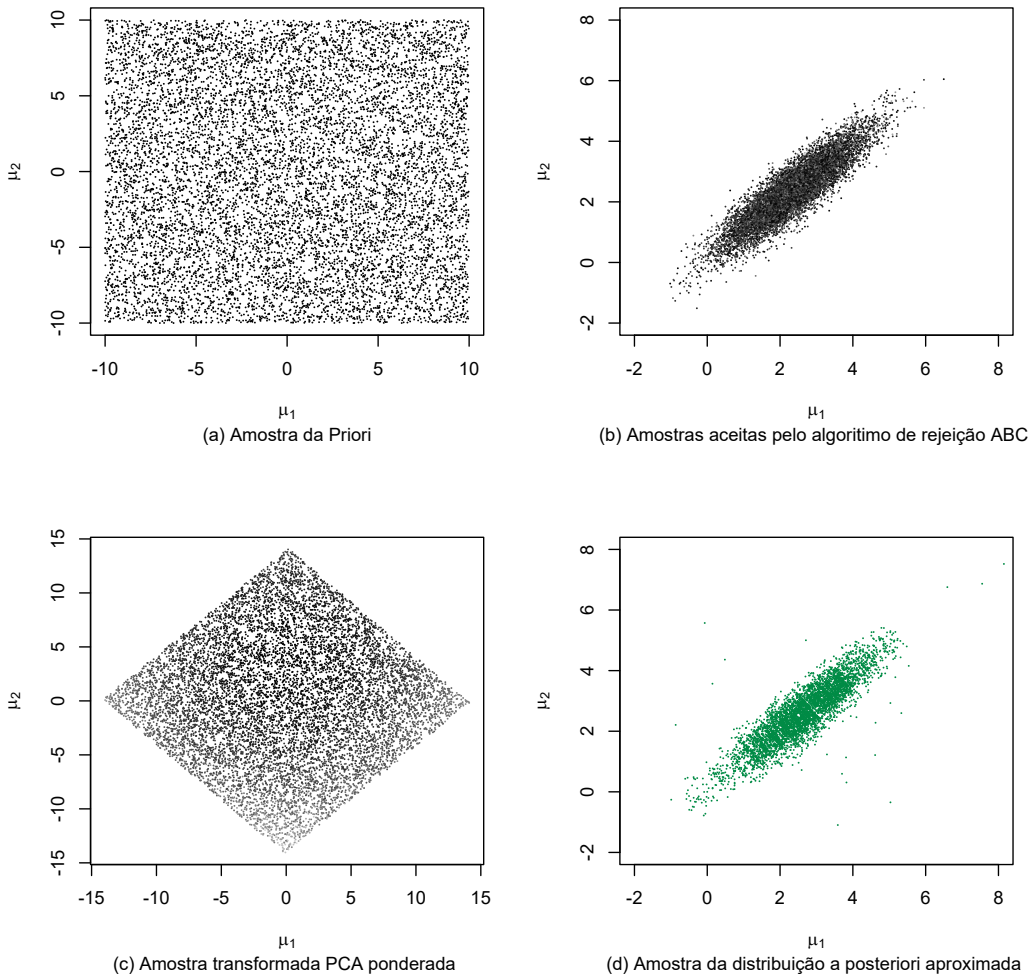



Figura 10: Etapas para estimação da distribuição a posteriori. (a) Distribuição a priori para $\boldsymbol{\mu}_1$ e $\boldsymbol{\mu}_2$. (b) Resultado da rodada preliminar do ABC por rejeição, para estimar os pesos das amostras. A intensidade de cinza representa o peso da respectiva amostra (quanto mais escuro maior o peso). (c) Descorrelação das amostras em (b) via PCA ponderada. (d) Estimativa da densidade posteriori baseada em 5 mil iterações do amostrador de Gibbs.

Percebe-se ainda o efeito da descorrelação na função de autocorrelação (FCA) Figura 11 (d), a FCA (referente a $\mu_1$) é insignificante em todos os *lags* para o método proposto, indicando que geramos efetivamente $m$ amostras independentes da posteriori. O mesmo não ocorre no método original, em que há uma forte correlação nos *lags* iniciais, com um decaimento exponencial (característico de um modelo auto regressivo). Obviamente, neste caso mais simples a própria estimativa do ABC já seria suficiente, conforme observa-se na Figura 10(b).

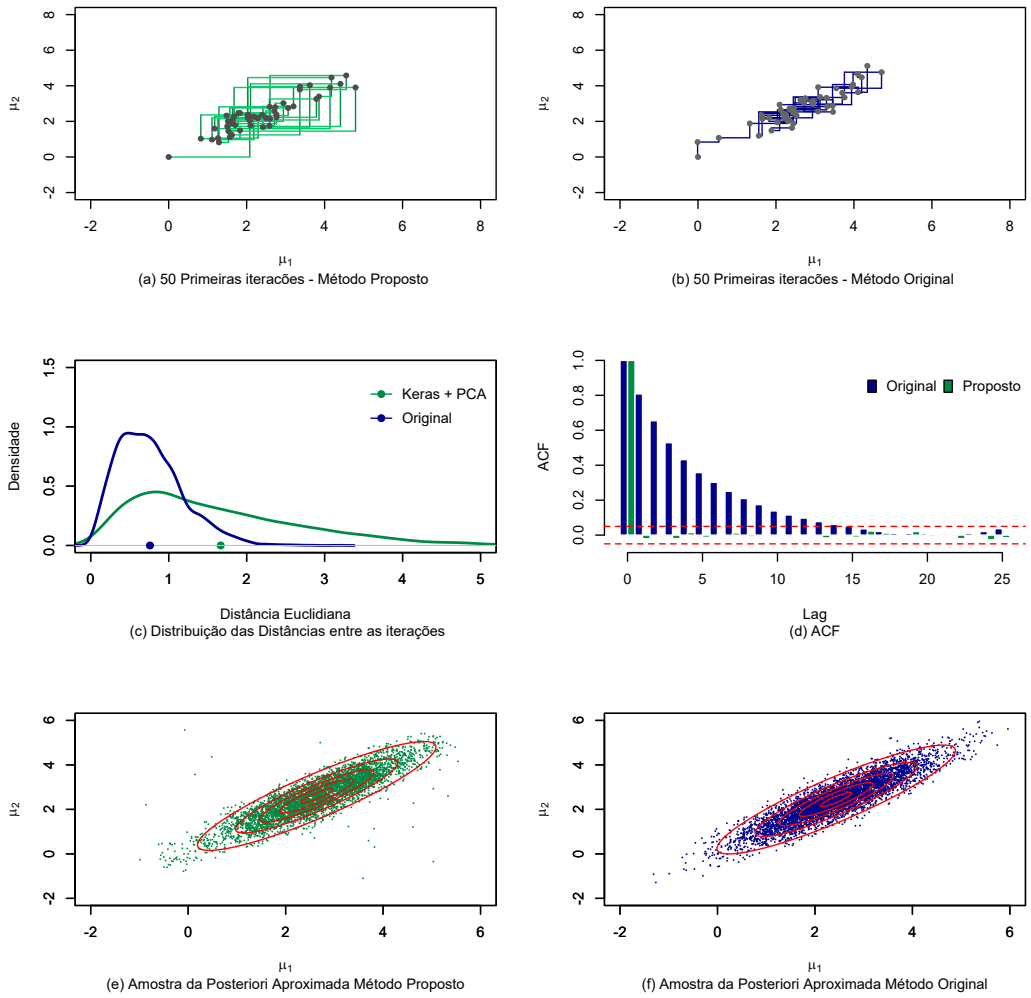



Figura 11: Comparativo do método original versus método proposto. (a) Primeiras 50 interações Gibbs - método proposto. (b) Primeiras 50 interações Gibbs - método original. (c) Distância Euclidiana entre pontos consecutivos da cadeia de Markov. (d) Função de autocorrelação. (e-f) Estimativa da densidade a posteriori baseada em 5 mil iterações do amostrador de Gibbs para o método original e proposto. Os contornos em vermelho representam as curvas de nível da distribuição a posteriori exata.

### 4.1.1 Custo Computacional

Para avaliar o efeito da etapa de descorrelação no tempo computacional gasto comparamos o tempo total gasto com e sem inclusão desta etapa. A partir da Tabela 1, verifica-se que, nesse exemplo, não há um aumento significativo no tempo gasto com a inclusão da etapa de descorrelação no método original (8,02 segundos) em relação ao método sem a etapa de descorrelação (7,45 segundos) usando modelos lineares.

| | Sem Descorrelação | | Com Descorrelação | |
|---|---|---|---|---|
| Etapa | Tempo (s) | % Tempo | Tempo (s) | % Tempo |
| Simulação de Dados Sintéticos | 6,56 | 88,06 | 6,56 | 81,80 |
| Descorrelação PCA | - | - | 0,74 | 9,23 |
| Estimação dos Modelos | 0,20 | 2,68 | 0,08 | 1,00 |
| Aproximação Gibbs Sampling | 0,69 | 9,26 | 0,64 | 7,97 |
| Total | 7,45 | 100,0 | 8,02 | 100,0 |

Tabela 1: Comparação do custo computacional para método original com e sem a etapa de descorrelação. Dados obtidos a partir do primeiro estudo de simulação item 4.1.

## 4.2 Exemplo 2: Modelo de Mistura Gaussiano

Consideramos neste exemplo o modelo de mistura gaussiano D-dimensional apresentado por Rodrigues, Nott e Sisson (2019) no qual a verossimilhança é dada por:

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

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, $\boldsymbol{\mu}(\boldsymbol{b},\boldsymbol{\theta}) = ((1-2b_1)\theta_1, \ldots, (1-2b_D)\theta_D)^\top$, $\boldsymbol{b} = (b_1, \ldots, b_D)^\top$ com $b_i \in \{0,1\}$ e $\boldsymbol{\Sigma} = [\Sigma_{ij}]$ tal que $\Sigma_{ii} = 1$ e $\Sigma_{ij} = \rho$ para $i \neq j$.

Para efeito de comparação, manteve-se os mesmos parâmetros utilizados por Rodrigues, Nott e Sisson (2019) quais sejam, $D = 2$, $\boldsymbol{s}_{obs} = (5/2, 5/2)$ e $w = 0,3$ e $\rho = 0,7$. A priori utilizada no artigo original, $U[-20, 40]$ , se mostrou demasiado vaga para o ajuste das redes, e por isso, nessa simulação, consideramos um intervalo menor, a saber $U[-10, 10]$.

As DCC's para $\theta_1$ e $b_1$, são dadas respectivamente por

$$\theta_1 \mid (\theta_2, \boldsymbol{b}, \boldsymbol{s}) \sim N(\mu_{\theta_1}, \sqrt{1-\rho^2})\ \mathbb{1}_{[-20,40]}(\theta_1),$$
$$\mu_{\theta_1} = s_1 - \rho s_2 + \rho\theta_2 - 2s_1 b_1 + 2\rho s_2 b_1 - 2\rho b_1\theta_2 - 2\rho\theta_2 b_2 + 4\rho b_1 b_2 \theta_2$$

$$b_1 \mid (\boldsymbol{\theta}, b_2, \boldsymbol{s}) \sim \text{Bernoulli}(L(p_{b_1})),$$
$$p_{b_1} = \ln\left(\frac{1-\omega}{\omega}\right) - \frac{2}{1-\rho^2}s_1\theta_1 + \frac{2\rho}{1-\rho^2}s_2\theta_1 - \frac{2\rho}{1-\rho^2}\theta_1\theta_2 + \frac{4\rho}{1-\rho^2}b_2\theta_1\theta_2,$$

em que $L(x) = 1/(1+\exp(-x))$ denota a função logística.

A distribuição de $\theta_1$ não é gaussiana, mas proporcional à gaussiana indicada no intervalo em que a priori é definida. As distribuições condicionais completas para $\theta_2$ e $b_2$ são obtidas de maneira similar.

Para obter $(\theta_1^i, \theta_2^i, b_1^i, b_2^i, s_1^i, s_2^i)$, $i = 1, ..., N$, geramos $N = 1.000.000$ de amostras de todo o processo, considerando $b(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})$. Posteriormente, para modelar as distribuições condicionais de $\theta_1$, $\theta_2$, ajustamos regressões quantílicas via redes neurais com correção *spline* monotônica. Para tanto, adotamos uma rede com 4 camadas intermediárias com 128, 64, 32, 16 neurônios respectivamente. Usamos função de ativação *relu*, *batchsize* igual a 32 e 30 épocas. Para os demais hiperparâmetros foram considerados os valores *defaut* do pacote *Keras*. Geramos $M = 20.000$ iterações utilizando como valor inicial $(\theta_1, \theta_2) = (0, -10)$.

Neste exemplo, inicialmente, testamos o uso do método proposto com a etapa de descorrelação, no entanto, ela se mostrou inviável, visto que houve uma piora na estimativa da distribuição a posteriori, pois a transformação via PCA mudou a geometria da função, dificultando a estimativa. Ou seja, neste caso, o uso da descorrelação dos parâmetros gerou uma perda real na metodologia.

Dado o exposto, procedemos com o ajuste das regressões quantílicas (quantis no intervalo [0,002 ; 0,998]) via redes neurais com correção *spline* monotônica sem realizar a etapa de descorrelação dos parâmetros.

Numa situação prática deve-se optar pela descorrelação quando a mistura da cadeia se mostrar lenta e os parâmetros foram altamente correlacionados à posteriori.

A Figura 12, que apresenta os resultados obtidos a partir do método proposto e original, sugere visualmente que a aproximação dos dois métodos foi parecida. Além disso, ambas são razoáveis, o que pode ser evidenciado na Figura 13(e) (onde temos os pontos representando a

distribuição a posteriori aproximada e os contornos representando a verdadeira distribuição a posteriori).

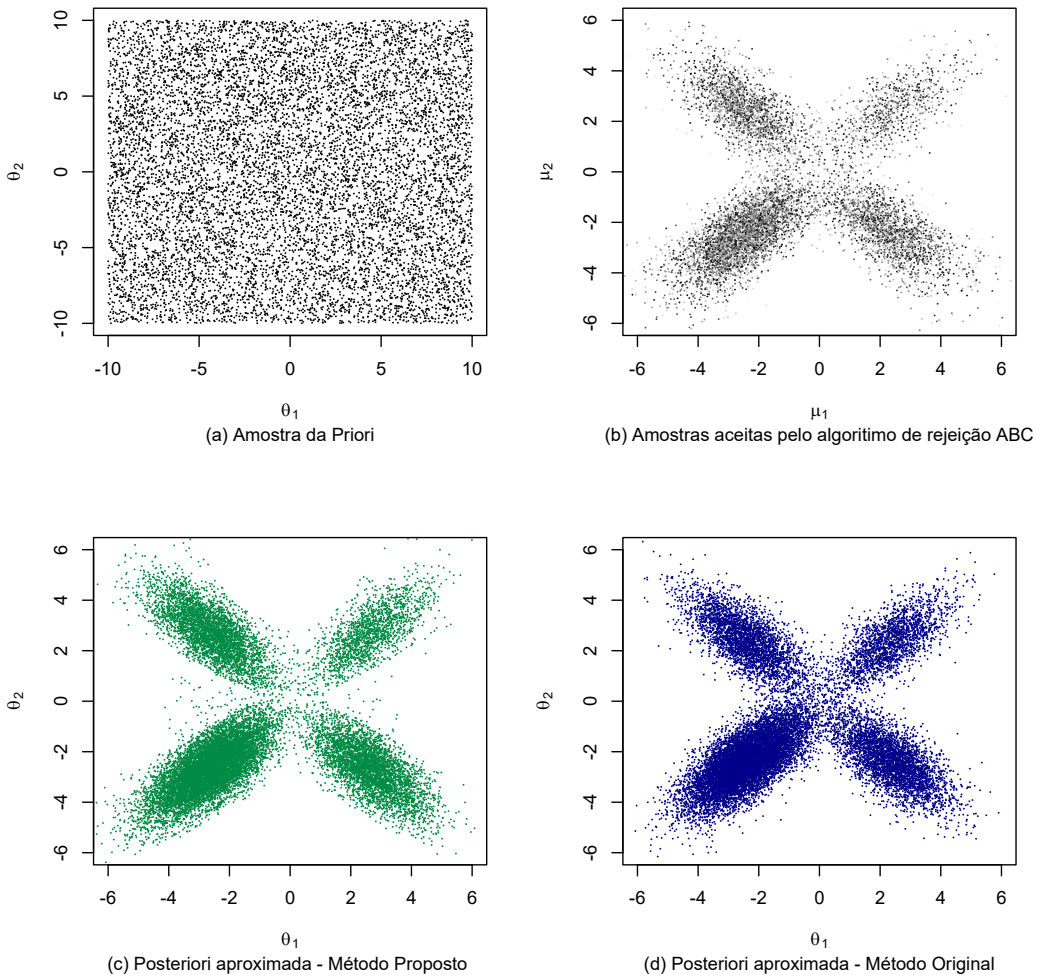



Figura 12: Etapas para estimação da distribuição a posteriori. (a) Distribuição a priori para $\boldsymbol{\theta}_1$ e $\boldsymbol{\theta}_2$. (b) Resultado da rodada preliminar do ABC por importância, para estimar os pesos das amostras. (c-d) Estimativa da densidade a posteriori baseada em 20 mil iterações do amostrador de Gibbs para os métodos proposto e original.

Neste exemplo, novamente, para avaliar a efetividade do método proposto em relação ao método apresentado por Rodrigues, Nott e Sisson (2019) comparamos os resultados da simulação realizada a partir da metodologia original com os resultados da metodologia proposta. Através da Figura 13 (a) - (b), ao analisar as cinquenta primeiras iterações de Gibbs, percebe-se que a mistura na cadeia para o método original é mais lenta que para o método proposto, o que pode ser observado através das distâncias entre os pontos, no qual nota-se que o intervalo de variação para as distribuição das distâncias entre os pontos do método proposto é maior que a do método original, Figura 13 (c) desta forma, pode-se dizer que houve um aumento na velocidade

da mistura na cadeia.

Na Figura 13 (d) observa-se, que mesmo sem optar pela etapa de descorrelação dos parâmetros, a função de autocorrelação do método proposto é praticamente insignificante em todos os *lags*. O que não ocorre no método original, em que há uma forte correlação nos *lags* iniciais, com um leve decaimento.

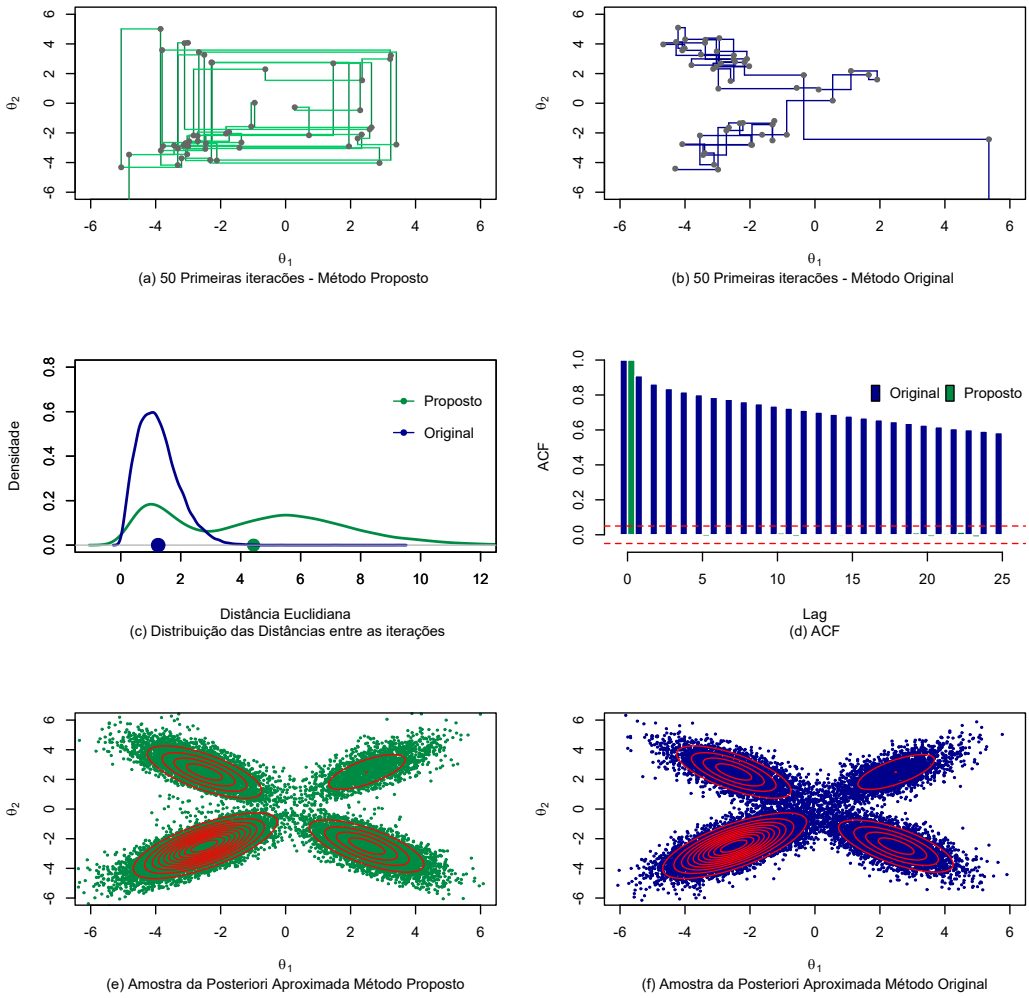
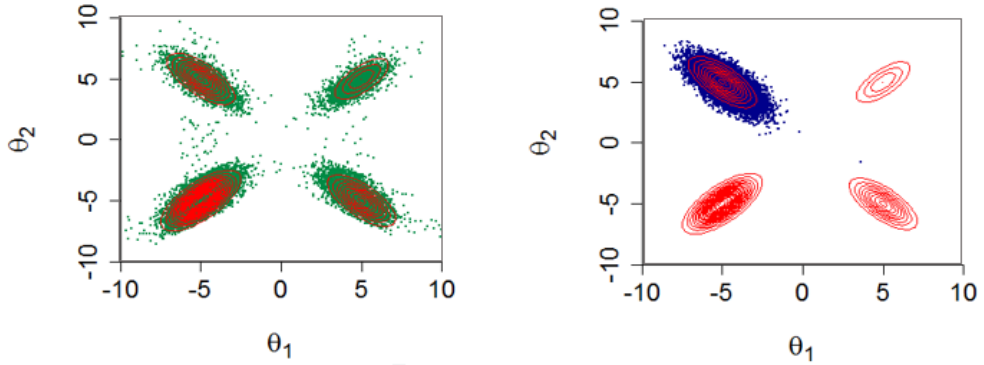



Figura 13: Comparativo do método original versus método proposto. (a-b) Primeiras 50 interações Gibbs para os métodos proposto e original. (c) Distância Euclidiana, entre os pontos, do método original versus proposto. (d) Função de autocorrelação. (e-f) Estimativa da densidade posteriori baseada em 20 mil iterações do amostrador de Gibbs para os métodos original e proposto e o contorno representando a verdadeira distribuição a posteriori .

Um dos motivos do melhor desempenho apresentado pelo método proposto frente ao original é que ao utilizar modelos flexíveis para estimação das condicionais completas não é necessário expandir o espaço por meio da utilização de variáveis latentes $b_1$ e $b_2$, para torná-las mais simples(o que nem sempre sempre é possível), como no método original.

Para exemplificar a vantagem do método proposto frente ao original , replicamos o exemplo 2 considerando os clusters mais separados, ou seja, $s_{obs} = (5, 5)$. A Figura 14 apresenta os resultados obtidos.




(a) Amostra da Posteriori Aproximada Método Proposto

(b) Amostra da Posteriori Aproximada Método Original

Figura 14: Estimativa da densidade posteriori baseada em 20 mil iterações do amostrador de Gibbs para os métodos proposto e original e o contorno representando a verdadeira distribuição a posteriori.

A partir dos resultados apresentados na Figura 14, verifica-se que ao cosiderar os clusters mais separados $s_{obs} = 5$ o método original não consegue explorar toda região ficando restrito a apenas um cluster. No entanto, dada a flexibilidade do método proposto isso não ocorre, mesmo para clusters mais distantes o método consegue explorar toda região. Ou seja o método proposto apresenta um ganho em termos da recuperação da posteriori.

E importante salientar que no método proposto, as etapas de estimação dos modelos quantílicos e de execução do amostrador de Gibbs aproximado apresentaram os maiores custos computacionais, representando 11,6% e 87,6%, respectivamente, do tempo total. Nos exemplos apresentados, na etapa de execução do amostrador de Gibbs a predição dominou o custo computacional, representando aproximadamente 100% do tempo. A interpolação via spline e a descorrelação dos parâmetros apresentaram custo computacional desprezível. A distribuição do custo computacional depende drasticamente do problema em estudo.

# 5 Conclusões

Neste trabalho introduzimos duas inovações para tornar mais eficiente o amostrador Gibbs-ABC apresentado por Rodrigues, Nott e Sisson (2019), o qual, assim como sua versão exata, pode apresentar lentidão na mistura da cadeia de Markov, sobretudo em modelos com fortes

correlações à posteriori.

Primeiramente, recomendamos realizar uma rodada preliminar de ABC para estimar a estrutura de correlação linear da distribuição a posteriori, a partir da qual podemos transformar os parâmetros via PCA, com o objetivo de reduzir as correlações existentes. Dessa maneira, é possível acelerar a convergência do amostrador de Gibbs sem que seja necessário aumentar o número de amostras sintéticas.

Em sua formulação original, o amostrador Gibbs-ABC aproxima as distribuições condicionais completas a partir de modelos ajustados para a média e (opcionalmente) para a variância condicional das distribuições. Dessa forma, ainda que os modelos adotados sejam flexíveis (de redes neurais, por exemplo), há uma suposição implícita de que a forma da distribuição é fixa em uma vizinhança de $s_{obs}$. Tal limitação estrutural dificulta, ou mesmo impossibilita, o alcance de bons resultados. Para mitigar esse problema, pode-se ajustar modelos localmente adequados em cada iteração de Gibbs. Entretanto, tal solução exige que os modelos de regressão sejam ajustados milhares de vezes, o que se mostra computacionalmente inviável, exceto para as construções mais simples.

Nesse contexto, propomos aproximar as DCC acumuladas por meio do ajuste de modelos de regressão quantílica via redes neurais com correção *spline* monotônica. Ao trabalhar com modelos flexíveis para os quantis, ao invés de para as médias e variâncias, conseguimos melhores aproximações para as DCCs, reduzindo, em última instância, o erro da estimativa da distribuição à posteriori de interesse. Sabidamente, redes neurais tendem a apresentar melhor desempenho quando são treinadas sobre grandes volumes de dados (por isso são ditas *data hungry*), dificultado portanto a aplicação da nossa técnica em modelos onde a geração de amostras sintéticas seja demasiado lenta.

O tempo computacional de nosso algoritmo é equivalente ao método original (com modelos para a média via redes neurais), visto que as etapas de interpolação via splines e de descorrelação dos parâmetros via PCA apresentam custo computacional desprezível.

Para avaliar empiricamente o desempenho do método proposto, foram realizados dois estudos de simulação: no primeiro, estimamos a distribuição a posteriori de um modelo normal bivariado. Verificamos que as duas inovações propostas para aprimorar o amostrador Gibbs-ABC apresentaram resultados satisfatórios — a velocidade de mistura da cadeia de Markov se mostrou drasticamente maior. De fato, o procedimento efetivamente eliminou a autocorrelação da cadeia.

No segundo estudo, estimamos a distribuição a posteriori de um modelo de mistura gaussiana bidimensional. Ao usar modelos de regressão quantílica, obtivemos bons resultados, mesmo sem expandir o vetor de parâmetros usando variáveis latentes (procedimento adotado com frequência para simplificar as DCCs). Esse exemplo mostra que nossa abordagem permite acomodar casos em que a geometria das condicionais completas seja substancialmente mais desafiadora.

Em suma, tendo em vista os resultados obtidos, entendemos que as inovações aqui propostas lograram êxito na busca por um amostrador de Gibbs-ABC mais eficiente. Obviamente, investigações comparativas mais extensas deverão ser conduzidas para estabelecer com mais precisão em quais situações nossa proposta deve ser empregada.

# Referências


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

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

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

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

Blum, M. G. B. et al. (2013). “A Comparative Review of Dimension Reduction Methods in Approximate Bayesian Computation”. *Statist. Sci.* 28, pp. 189–208.

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.

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

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

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

Faceli, K. et al. (2011). *Inteligência Artificial: Uma Abordagem de Aprendizagem de Máquina*. LTC.

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

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

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

Haykin, S. (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.

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

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

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

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

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

Koenker, R. (2005). *Quantile Regression*. Econometric Society Monographs. Cambridge University Press. ISBN: 9780521608275,0521608279. URL: http://gen.lib.rus.ec/book/index.php?md5=FEE3C74082DF706D923A9B3D104B1950.

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.

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

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.

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

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

Nott, D. J. et al. (2018a). *Approximation of Bayesian Predictive p-Values with Regression ABC*. Vol. 13. Bayesian Anal.

Nott, D. J. et al. (2018b). *High-dimensional approximate Bayesian computation*. arXiv:1802.09725 [stat.CO].

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

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

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.

Rodrigues, G. S., Nott, D. J. e Sisson, S. A. (2019). *Likelihood-free approximate Gibbs sampling*. arXiv:1906.04347.

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

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

Tavare, S. et al. (1997). *Inferring Coalescence Times From DNA Sequence Data*. Vol. 145. Genetics, pp. 505–518.

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.