Relatório Final

Introdução às Cadeias Estocásticas com Memória de Alcance Variável

Caio Théodore Genovese Huss Oliveira
Orientador: Christophe Frédéric Gallesco

Conteúdo

1 Identificação

Projeto "Introdução às Cadeias Estocásticas com Memória de Alcance Variável", realizado pelo aluno Caio Théodore Genovese Huss Oliveira, RA 247005, com a orientação do professor Christophe Frédéric Gallesco, de setembro de 2024 a julho de 2025, no Instituto de Matemática, Estatística e Computação Científica.

2 Introdução

2.1 Objetivos do projeto

Durante a primeira parte do projeto, o aluno estudou os fundamentos teóricos das Cadeias Estocásticas com Memória de Alcance Variável (CEMAV). Em particular, as noções essenciais de *contexto* e *árvore probabilística de contexto* foram abordadas. A questão da existência e unicidade de processos estacionários compatíveis com uma dada árvore probabilística de contexto também foi estudada. Em seguida, o aluno se familiarizou com o *algoritmo de contexto* introduzido por Rissanen, permitindo a estimativa de forma consistente tanto do tamanho do contexto quanto das probabilidades de transição associadas.

Na segunda parte do projeto, o aluno buscaria na literatura aplicações de CEMAV para modelagem de problemas concretos. Sob reserva de disponibilidade de dados, o aluno poderia enfim desenvolver códigos em R e/ou Python para testar modelos de CEMAV.

A segunda parte do projeto passou a envolver dados simulados pelo próprio aluno, o estudo do conceito de *esqueleto*, a elaboração de um algoritmo para a extração do *esqueleto* de uma cadeia de Markov, e a implementação do mesmo em R, resultado da nova pesquisa do orientador e dada a possibilidade de publicar um artigo.

2.2 Resumo de resultados

O conceito de CEMAV foi estudado profundamente, por meio tanto de fontes teóricas quanto de simulações do próprio aluno em R, envolvendo CEMAVs e cadeias de ordem infinita. Construiu-se um entendimento forte sobre o algoritmo de contexto. Provas teóricas foram replicadas sobre CEMAVs.

Dois algoritmos de extração de *esqueleto* foram obtidos, um a partir da árvore probabilística, outro a partir de dados reais, permitindo uma redução de custo para a prova de propriedades, como a irredutibilidade e a periodicidade. Em alguns casos, esta redução de custo é aproximadamente cúbica. O algoritmo empírico foi implementado em um pacote de R.

2.3 Cadeias Estocásticas com Memória de Alcance Variável

CEMAVs foram introduzidas por Jorma Rissanen, em 1983 [Ris83], como um método de compressão de dados binários. A diferença de uma CEMAV para uma cadeia estocástica tradicional está na variação do tamanho (ou alcance) da memória associada às probabilidades do próximo estado (ou símbolo). Uma CEMAV de ordem k possui passados relevantes às probabilidades, chamados de contextos, de tamanho menor ou igual a k. Uma definição rigorosa se encontra na seção 4.1.

Em seu artigo, Rissanen visualiza estes contextos na forma de uma árvore probabilística, em que cada folha (e o caminho até ela) representa um contexto da CEMAV, lido da raíz à folha (do presente ao passado, ou da direita à esquerda na sequência). O formato foi adotado na literatura e é utilizado neste relatório.

O número de contextos possíveis de uma cadeia tradicional cresce exponencialmente à medida que se aumenta a ordem da mesma: Sendo $\mathcal{A} = \{a_1, a_2, \cdots, a_{|\mathcal{A}|}\}$ o conjunto de símbolos possíveis de uma cadeia (alfabeto), uma cadeia de ordem k possui $|\mathcal{A}|^k$ passados relevantes diferentes. CEMAVs permitem capturar passados relevantes extensos sem aumentar o tamanho de toda a cadeia, economizando recursos consideráveis, além de observar relações entre os símbolos na sequência e o tamanho do passado relevante na mesma.

Após o artigo inicial de Rissanen, CEMAVs encontraram aplicações multidisciplinares, entre as quais:

- **Linguística:** CEMAVs podem modelar diferentes estruturas rítmicas na linguagem, permitindo, por exemplo, mapear diferenças no ritmo de texto entre o português brasileiro e o de Portugal [Gal+12].
- **Microbiologia:** Ganhos de eficiência e seleção de ordem quando comparadas a cadeias de ordem fixa, para o sequenciamento de DNA, ajudando a entender o comportamento de comunidades microbiais [Lia+16].
- **Web:** Podem ser utilizadas para modelar e prever o comportamento de um usuário na internet [BL07].

2.4 Esqueletos

Uma redução de ordem de cadeias de ordem infinita foi desenvolvida em 2025 por Christophe Gallesco, Alexsandro Gallo, e Daniel Takahashi [GGT25]. A noção de *esqueleto* de uma cadeia estocástica é um caso particular deste novo resultado, e será publicada posteriormente em um artigo co-autorado pelo aluno orientado e os professores Christophe Gallesco e Daniel Takahashi [GOT25], que também fornece um algoritmo para obter o *esqueleto* a partir de uma cadeia de Markov.

O *esqueleto* limita-se ao passado necessário para determinar se uma probabilidade é nula, possuindo ordem menor que a da cadeia em muitos casos, porém mantendo propriedades como a irredutibilidade e periodicidade [GOT25]. Como será detalhado mais a frente, a redução de ordem decorrente do *esqueleto* permite uma diminuição expressiva nos custos computacionais da análise destas propriedades. Em casos que a ordem da cadeia é consideravelmente maior que a do *esqueleto*, esta redução de custo torna-se aproximadamente cúbica (demonstrado na seção 5.2).

Em adição ao artigo a ser publicado com o professor orientador, um algoritmo empírico foi desenvolvido, permitindo obter o *esqueleto* de uma cadeia a partir de dados. Uma implementação do algoritmo empírico também foi desenvolvida na linguagem R, e incorporada no pacote *skeleton*. O algoritmo e o pacote em R serão publicados futuramente pelo aluno orientado.

3 Materiais, Métodos e Cronograma

Todos os códigos foram desenvolvidos na linguagem R [R24]. Os códigos para a estimação do *esqueleto* foram em parte inspirados pelo pacote *VLMC* [Mae24], e otimizados com base nas práticas do livro *Advanced R* [Sau20].

3.1 Cronograma

Incluindo atividades previstas e incluídas posteriormente:

1. **Estudos e exercícios teóricos** - Setembro de 2024 a Abril de 2025 - Revisão da literatura e familiarização com os conceitos do projeto. Exercícios e repetição de provas teóricas.

2. **Simulações iniciais de CEMAVs** - Outubro a Dezembro de 2024 - Simulações de CEMAVs e cadeias de ordem infinita binárias, complementando os estudos teóricos e permitindo visualizar propriedades.

3. **Desenvolvimento de algoritmos** - Fevereiro a Junho de 2025 - Construção e modificação dos algoritmos presentes na seção 4.

4. **Criação e Simulação de CEMAVs-teste** - Fevereiro a Maio de 2025 - Simulação das CEMAVs utilizadas para o desenvolvimento e validação da implementação do algoritmo de extração do *esqueleto*.

5. **Elaboração do Relatório Parcial** - Fevereiro a Março de 2025

6. **Implementação do algoritmo em R** - Março a Junho de 2025 - Implementação do algoritmo em R, ajustes no código para envolver cadeias com $|\mathcal{A}| > 2$, e desenvolvimento do pacote de R (pacote ainda em desenvolvimento).

7. **Elaboração do Relatório Final** - Julho de 2025

8. **Artigo** - Junho de 2025, em andamento - Artigo a ser publicado com o orientador e outro professor, sobre o algoritmo de extração do *esqueleto*, ainda não concluído.

4 Resultados

4.1 Resultados Teóricos

Notação

Seja $\mathcal{A} = \{a_1, a_2, \cdots, a_{|\mathcal{A}|}\}$ um alfabeto finito de tamanho $|\mathcal{A}|$. Representamos por x_i^j uma sequência $(x_i, \cdots, x_j)$ de símbolos em $\mathcal{A}$, com a convenção $x_i^j = \varnothing$ se $i > j$. Seja $\mathcal{A}_+^* = \bigcup_{k=1}^{\infty} \mathcal{A}^{\{-k,\cdots,-1\}}$ o conjunto de todas as sequências finitas, e $\mathcal{A}_- = \mathcal{A}^{\{\cdots,-n,\cdots,-2,-1\}}$ o conjunto de passados infinitos, $x_{-\infty}^{-1} \in \mathcal{A}_-$.

4.1.1 CEMAV

Definição 4.1. A **função de alcance de contexto**, é uma função $l : \mathcal{A}_+^* \to \mathbb{N} \cup \{\infty\}$, que satisfaz:

a) $\forall k \geq 1, x_{-k}^{-1} \in \mathcal{A}_+^*,\ l(x_{-k}^{-1}) \in \{1, \cdots, k\} \cup \{\infty\}$.

b) $\forall x_{-\infty}^{-1} \in \mathcal{A}_-, \exists k \in \mathbb{N}$ tal que $l(x_{-k}^{-1}) = k \Rightarrow l(x_{-i}^{-1}) = \begin{cases} \infty, i < k. \\ k, i \geq k. \end{cases}$

Definimos então, para $x_{-\infty}^{-1} \in \mathcal{A}_-$: $l(x_{-\infty}^{-1}) = \{\inf k \geq 1 : l(x_{-k}^{-1}) \in \mathbb{N}\}$, $\inf \varnothing = +\infty$.

Definição 4.2. Para uma sequência infinita $x_{-\infty}^{-1} \in \mathcal{A}_-$, o **contexto** associado à função $l(\cdot)$ é $x_{-l(x_{-\infty}^{-1})}^{-1}$.

Definição 4.3. Seja $l(\cdot)$ uma função de alcance de contexto. Uma cadeia estocástica $(X_n)_{n\in\mathbb{Z}}$ com valores em $\mathcal{A}$ é uma **Cadeia Estocástica com Memória de Alcance Variável (CEMAV)** se, para todo passado $x_{-\infty}^{-1} \in \mathcal{A}_-$ e todo símbolo $a \in \mathcal{A}$:

$$\mathbb{P}[X_0 = a \,|\, X_{-\infty}^{-1} = x_{-\infty}^{-1}] = \mathbb{P}[X_0 = a \,|\, X_{-l(x_{-\infty}^{-1})}^{-1} = x_{-l(x_{-\infty}^{-1})}^{-1}].$$

Simplificando a notação, com $k = l(x_{-\infty}^{-1})$, definimos,

$$p(a \,|\, x_{-k}^{-1}) = \mathbb{P}[X_0 = a \,|\, X_{-k}^{-1} = x_{-k}^{-1}]. \tag{1}$$

Definição 4.4. Uma sequência $x_{-k}^{-1} \in \mathcal{A}_+^*$ possui como **sufixos** as sequências $x_{-j}^{-1}, j \leq k$. x_{-j}^{-1} é *sufixo próprio* de x_{-k}^{-1} se $j < k$. Um conjunto $\mathcal{S} \subset \mathcal{A}_+^*$ satisfaz a *propriedade de suffixo* se nenhum dos elementos de $\mathcal{S}$ for *sufixo próprio* de outro.

Definição 4.5. Uma **árvore probabilística de contextos** (chamada de árvore ou árvore de contextos neste relatório, por brevidade) em $\mathcal{A}$ é composta dos pares (τ, p), onde:

- $\tau = \tau^{l(\cdot)} = \{x_{-k}^{-1} : k = l(x_{-k}^{-1}), k \geq 1\}$ para uma função de alcance de contexto $l(\cdot)$.
- $p = \{p(\cdot \,|\, \underline{x}), \underline{x} \in \tau\}$.

A possibilidade de representar τ no formato de árvore vem do fato que τ satisfaz a *propriedade de sufixo* (decorrente da Definição 4.1).

Uma cadeia estocástica $(X_n)_{n\in\mathbb{Z}}$ é compatível com uma árvore (τ, p) se (1) vale para todo $\underline{x} \in \tau$. A cadeia possui *ordem* $k = \sup_{x\in\tau} l(x)$. Caso $k = +\infty$, é uma cadeia de ordem infinita. Este conjunto de definições estabelece rigorosamente o conceito de CEMAV utilizado neste projeto. Mais detalhes podem ser encontrados no trabalho de A. Galvez e E. Löcherbach [GL08].

4.1.2 Algoritmo de Contexto

J. Rissanen apresentou, junto com a ideia de CEMAV, um algoritmo para extraí-las no caso binário $\mathcal{A} = \{0, 1\}$ [Ris83]. Sendo $X_0, X_1, \cdots, X_{n-1}$ amostras da árvore (τ, p), z um passado finito, e s um sequência, ambos compostos por símbolos em $\mathcal{A}$, consideramos:

A contagem de ocorrências de s, $c(s) = \sum_{t=0}^{n-j} \mathbf{1}\{X_t^{t+j} = s\}$, $c(\varnothing) = n$, os estimadores de probabilidade $p(a \,|\, z) = c(za)/c(z)$, $p(z) = c(z)/c(\varnothing)$, a *entropia* $H(\mathcal{A}, z) = -p \log p - (1-p)\log(1-p)$, $p = p(0 \,|\, z)$, e o *incremento condicional de entropia* $\Delta(t, z) = p(z)H(\mathcal{A}, z) - p(0z)H(\mathcal{A}, 0z) - p(1z)H(\mathcal{A}, 1z)$.

Para um símbolo observado x_t, escolhemos seu *contexto* $z^*(t)$ como sendo o maior sufixo z do passado $(x_0, \cdots, x_{t-1})$ tal que:

I) $\Delta(t, z) > t^{-1} log(t)$,

II) $|z| \leq \beta \log t$,

III) $\min\{c(0z), c(1z)\} \geq 2\frac{\alpha t}{\sqrt{\log t}}$,

onde α, β são constantes positivas. A escolha de um contexto $z^*(v)$ que tenha $z^*(t)$ como sufixo faz com que $z^*(t)$ deixe de ser contexto.

Obtemos o conjunto de contextos $Z = \bigcup_{t=0}^{n-1} z^*(t)$ que, por respeitar a propriedade de sufixo, pode ser representado como uma árvore de contexto, em conjunto com os estimadores de probabilidade $p(a \,|\, z^*), a \in \mathcal{A}, z^* \in Z$. O resultado principal do artigo de Rissanen é que, sendo Z^0 o conjunto de contextos da cadeia estocástica geradora dos dados,

$$\lim_{n\to\infty} \mathbb{P}\{z^*(n) \in Z^0\} = 1.$$

4.1.3 Esqueleto

O principal resultado desta iniciação científica é a criação e implementação de um algoritmo para extrair o *esqueleto* de uma cadeia estocástica. Portanto, é importante fornecer uma definição mais rigorosa deste conceito. O *esqueleto* é um caso especial das ideias desenvolvidas em [GGT25]. Definimos primeiro a operação de concatenação. Sejam $x = x_1^j = (x_1, \cdots, x_j),\ y = y_{j+1}^k = (y_{j+1}, \cdots, y_k)$,

$$z = z_1^k = xy \Leftrightarrow z = (x_1, \cdots, x_j, y_{j+1}, \cdots, y_k),\ z \in \mathcal{A}^k, x \in \mathcal{A}^j, y \in \mathcal{A}^{k-j}.$$

A operação segue a convenção de que $\varnothing x = x\varnothing = x$.

Seja $(X_n)_{n\in\mathbb{Z}}$ uma cadeia de markov com ordem $m \geq 1$,

Definição 4.6. $\forall a \in \mathcal{A}, x \in \mathcal{A}^m$, definimos

$$\tau(x,a) = \min\{i \geq 0 : p(a \,|\, yx_{-i}^{-1}) > 0, \forall y \in \mathcal{A}_- \text{ ou } p(a \,|\, yx_{-i}^{-1}) = 0, \forall y \in \mathcal{A}_-\}.$$

$$\tau_x = \sup_{a\in\mathcal{A}} \tau(x,a).$$

O conjunto de contextos $\mathcal{S} = \bigcup_{x\in\mathcal{A}^m} \{x_{-\tau_x}^{-1}\}$ é chamado de **esqueleto** da cadeia, possuindo ordem $d = \sup_x \tau_x \leq m$.

Como mencionado na introdução, o *esqueleto* compartilha as propriedades de irredutibilidade e periodicidade com a cadeia [GOT25].

A análise da irredutibilidade de uma cadeia de Markov de ordem 1, com alfabeto $\mathcal{A}$ de tamanho $|\mathcal{A}|$ envolve $|\mathcal{A}|-1$ multiplicações de uma matriz $M_{|\mathcal{A}|\times|\mathcal{A}|}$, verificando se, para todo par $(u,v) \in \mathcal{A}^2, \exists n \in \mathbb{N}, t.q.\ M^n > 0$. Para uma cadeia de ordem $m \geq 1$, ou um *esqueleto* de ordem $d \geq 1$, podemos considerá-los como cadeias de ordem 1 com alfabeto $\mathcal{A}^m$ e $\mathcal{A}^d$ [GOT25], respectivamente, e seguir com a análise da irredutibilidade de forma similar.

A análise da irredutibilidade do esqueleto pode frequentemente ser acelerada, usando

Lema 1. *Considere um esqueleto $\mathcal{S}$ de ordem d. A existência de um contexto $x \in \mathcal{S}$, t.q. $\tau_x < d$, para o qual $\exists a \in \mathcal{A}$, $p(a \,|\, x) = 0$ implica na não irredutibilidade do esqueleto.*

Para a prova, definimos primeiro a **matriz de transição do esqueleto**.

Definição 4.7. Considere um *esqueleto* de ordem d sobre $\mathcal{A}$. Sua matriz de transição, $\mathcal{Q} : \mathcal{A}^d \times \mathcal{A}^d \to \{0, 1\}$, é definida por:

$$\mathcal{Q}(i,j) = \begin{cases} 1, \ se\,(i,j) = (x^0_{-d+1}, x^0_{-d+2}a)\, e\, p(a \,|\, yx^0_{-d+1}) > 0, \ \forall y. \\ 0, \ \text{ caso contrário.} \end{cases}$$

Seguindo com a prova:

Demonstração. $p(a \,|\, x) = 0$, logo $\mathbb{P}[xa] = 0$, onde $\mathbb{P}[xa]$ é a probabilidade da sequência xa ocorrer na cadeia. $|x| < d \Rightarrow |xa| \leq d$, portanto xa é sufixo de pelo menos uma sequência $y \in \mathcal{A}^d$, logo $\exists y \in \mathcal{A}^d \, t.q. \, \mathbb{P}[y] = 0 \Rightarrow \mathcal{Q}(i,y) = 0, \forall i \in \mathcal{A}^d$ e, portanto, o *esqueleto* não é irredutível. □

4.1.4 Exercícios

Por último, exercícios sobre as propriedades de *continuidade, log-continuidade, e não-nulidade fraca e forte/estrita* de cadeias infinitas binárias ($\mathcal{A} = \{0, 1\}$), encontrados em [FGG01]:

Defina $l(x^{-1}_{-\infty}) = \inf_{i \in \mathbb{N}}\{i : x_{-i-1} = 1\}, l(0^\infty) = \infty, \ e \ q_k = p(1 \,|\, x^{-1}_{l(x^{-1}_{-\infty})})$, com $0 < q_k < 1$. As seguintes propriedades foram provadas:

- Se o limite $\lim_{k\to\infty} q_k$ não existir ou for diferente de q_∞, a cadeia não é contínua.
- Se $\lim_{k\to\infty} q_k = q_\infty, \exists 0 < c \leq d \leq 1$ tal que $q_k \in [c, d], \forall k$, então a cadeia é log-contínua e fortemente não nula.
- Se $\lim_{k\to\infty} q_k = q_\infty = 0$, a cadeia é contínua sem ser log-contínua, e possui não-nulidade apenas fraca.

4.2 Simulação de Cadeias

CEMAVs foram simuladas com códigos na linguagem R, afim de entender melhor seu comportamento a longo prazo. As cadeias seguem o exemplo providenciado em [GL08], sendo binárias ($\mathcal{A} = \{0, 1\}$), e com função de alcance do contexto:

$$l(x^{-1}_{-\infty}) = \inf_{i \in \mathbb{N}}\{i : x_{-i} = 1\}.$$

Dois casos foram explorados:

- $p(1 \mid 10^{k-1}) = \frac{1}{k} \Rightarrow \sum_{x \in \tau} p(1 \mid x) = +\infty$, quase certamente haverá um número infinito de 1s.[GL08] (Prop. 2.10, provada pelo aluno como exercício).
- $p(1 \mid 10^{k-1}) = \frac{1}{k^n}, n > 1 \Rightarrow \sum_{x \in \tau} p(1 \mid x) < \infty$, possibilidade de uma sequência infinita de 0s.

onde 10^k representa um símbolo 1 seguido de k símbolos 0.

4.2.1 Cadeias para desenvolvimento do algoritmo

Cadeia 1 é uma CEMAV binária de ordem $k = 10$, com $\mathcal{A} = \{0, 1\}$, cujo *esqueleto* possui ordem $d = 3$. Seus contextos foram decididos manualmente, e podem ser observados na árvore presenta na Figura 3. A cadeia possui duas transições proibidas,

$$p(1 \mid 10) = 0, \quad p(1 \mid 111) = 0,$$

resultando no *esqueleto* presente na árvore (Figura 3), e a matriz de transições de *esqueleto* na Figura 1. As probabilidades restantes foram definidas de forma aleatória: Seja $(w)_i$ a sequência dos contextos em ordem alfabética (por tamanho, e começando pelo passado mais recente). As probabilidades $p(1 \mid W_i) = \mathbb{P}[X_0 = 1 \mid X_{-l}^{-1} = W_i]$ são definidas por:

1. $p(W_1)$ e $p(W_2)$ são as transições proibidas do *esqueleto*.
2. $p(W_3) \sim N(0,3\,,\, 0,04)$.
3. $p(W_i) \sim N(1 - p(W_{i-1})\,,\, 0,04)$.
4. Probabilidades fora de $[0,06\,,\, 0,94]$ foram arredondadas para o limite mais próximo. Isto foi feito para evitar que probabilidades muito pequenas interferissem com o ajuste de modelo.

Cadeia 2 é uma cadeia de Markov com ordem $k = 5$, e *esqueleto* de ordem $d = 3$, representado na Figura 2. A árvore completa é grande demais para ser representada visualmente neste relatório. A cadeia possui alfabeto de tamanho 4 $\mathcal{A} = \{\text{sol, nuvem, chuva, tempestade}\}$, e transições proibidas:

$$p(chuva \mid sol\ sol) = 0,\ p(w \mid tempestade\ sol\ sol) = \begin{cases} 1,\ w = nuvem \\ 0,\ c.c. \end{cases}$$

$$p(nuvem \mid nuvem\ chuva) = p(tempestade \mid nuvem\ chuva) = 0$$

$$p(chuva \mid tempestade) = p(tempestade \mid tempestade) = 0.$$

Uma cadeia de ordem fixa foi preferida para explorar simulações automáticas: Uma função será implementada no pacote *skeleton*, com os códigos utilizados na simulação da *Cadeia 2*, para que possa se simular dados provenientes de uma cadeia, fornecendo apenas as transições proibidas do *esqueleto* e a ordem da cadeia desejada. A escolha de probabilidades da cadeia segue os passos:

1. Forneça a ordem da cadeia a ser simulada k, o limite de probabilidade c (0,06 na *Cadeia 1*), e os vetores contendo transições proíbidas $t(w) = (t_1(w), \cdots, t_{|\mathcal{A}|}(w))$, $t_i = \mathbf{1}\{p(a_i \,|\, w) > 0\}$, para contextos de *esqueleto* w.

2. Gere uma lista de (contextos | vetores) adicionando os sufixos necessários aos contextos w de *esqueleto* fornecidos para obter todo o conjunto $\mathcal{A}^k$. Para passados $s \in \mathcal{A}^k$ que não possuem nenhum dos contextos de *esqueleto* como sufixo, associe $t(s) = (1, \cdots, 1)$.

3. Seja $h = |\mathcal{A}|$, $|t(s)| = \sum_{i=1}^{h} t_i(s)$ e $\mathcal{A}^{t(s)} = \{a_i : t_i(s) = 1\}$. Para cada $s \in \mathcal{A}^k$, gere as probabilidades associadas a este passado:

$$p(a_i \,|\, s) : \begin{cases} = 0, \ se\, a_i \notin \mathcal{A}^{t(s)}. \\ \sim U(c, 1 - c[|t(s)| - \sum_{j=1}^{i} t_j(s)]), \ se\, \exists l > i,\ t_l(s) = 1. \\ = 1 - \sum_{j=1}^{i-1} p(a_j \,|\, s), \ se\ t_l(s) = 0,\ \forall l > i. \end{cases}$$

Cadeia 2 foi simulada com $c = 0,05$. Ambas cadeias foram simuladas construindo um passado inicial composto por $2k$ símbolos, por amostra aleatória com reposição, utilizando probabilidades $p(a) = \frac{1}{|\mathcal{A}|}, a \in \mathcal{A}$, e então simulando a cadeia a partir deste passado. Obteve-se uma amostra de 100 mil símbolos (excluindo o passado inicial) para cada cadeia, em formato de texto, disponível nos códigos.

Cadeia 1 e Cadeia 2 foram simuladas com o objetivo de validar a implementação do algoritmo de extração do esqueleto empírico (Seção 4.3.2), comparando os resultados obtidos com as árvores teóricas nas Figuras 2 e 3. *Cadeia 1* foi utilizada na fase inicial de desenvolvimento do algoritmo, e *Cadeia 2* validou a inclusão de alfabetos com $|\mathcal{A}| > 2$ e símbolos com mais de um caractere.

As *cadeias 1 e 2* possuem o mesmo número de possíveis passados relevantes, $2^{10} = 4^5 = 1024$, resultando em matrizes de transição de tamanho 1024×1024. As matrizes também são esparsas, já que, para cada linha, um máximo de $|\mathcal{A}| = 2$ (*Cadeia 1*) ou 4 (*Cadeia 2*) colunas representam uma transição com probabilidade positiva, de 1024. Por esses motivos, não há uma visualizaçõ interessantes das matrizes de transição originais.

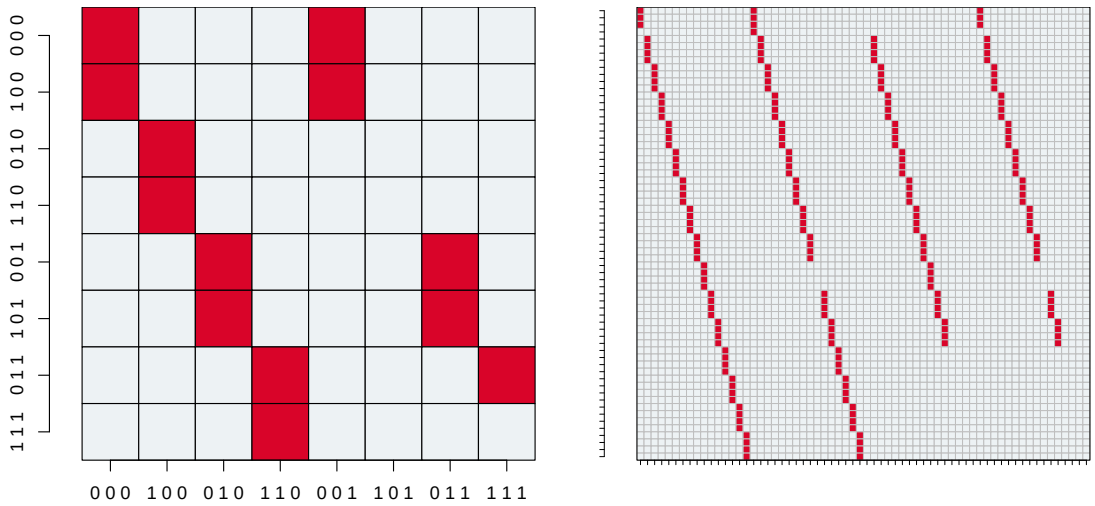

Figura 1: Matrizes de transição dos esqueletos das *Cadeias 1* (esquerda, tamanho 8×8) e *2* (direita, tamanho 64×64). Transições com probabilidade positiva em vermelho.

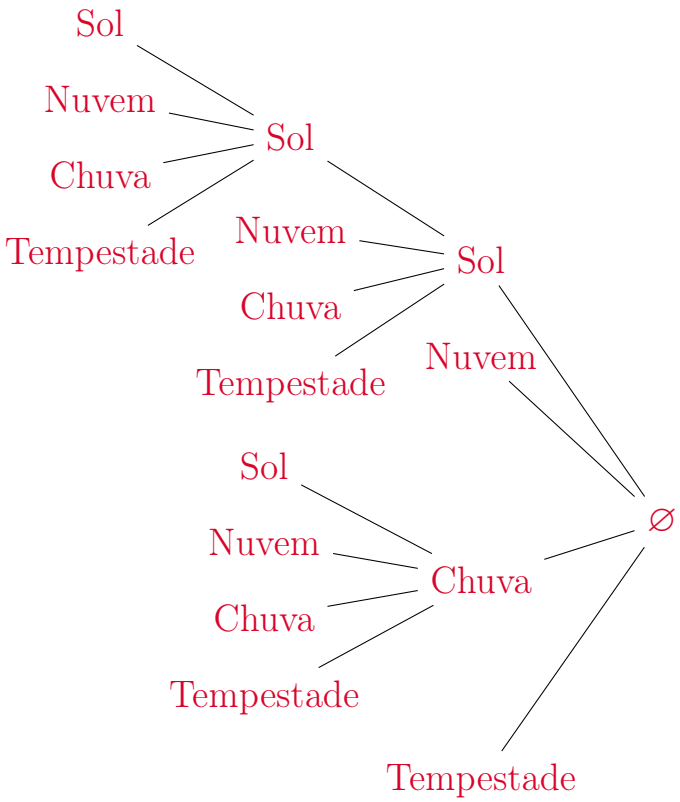

Figura 2: Árvore do esqueleto de *Cadeia 2*.

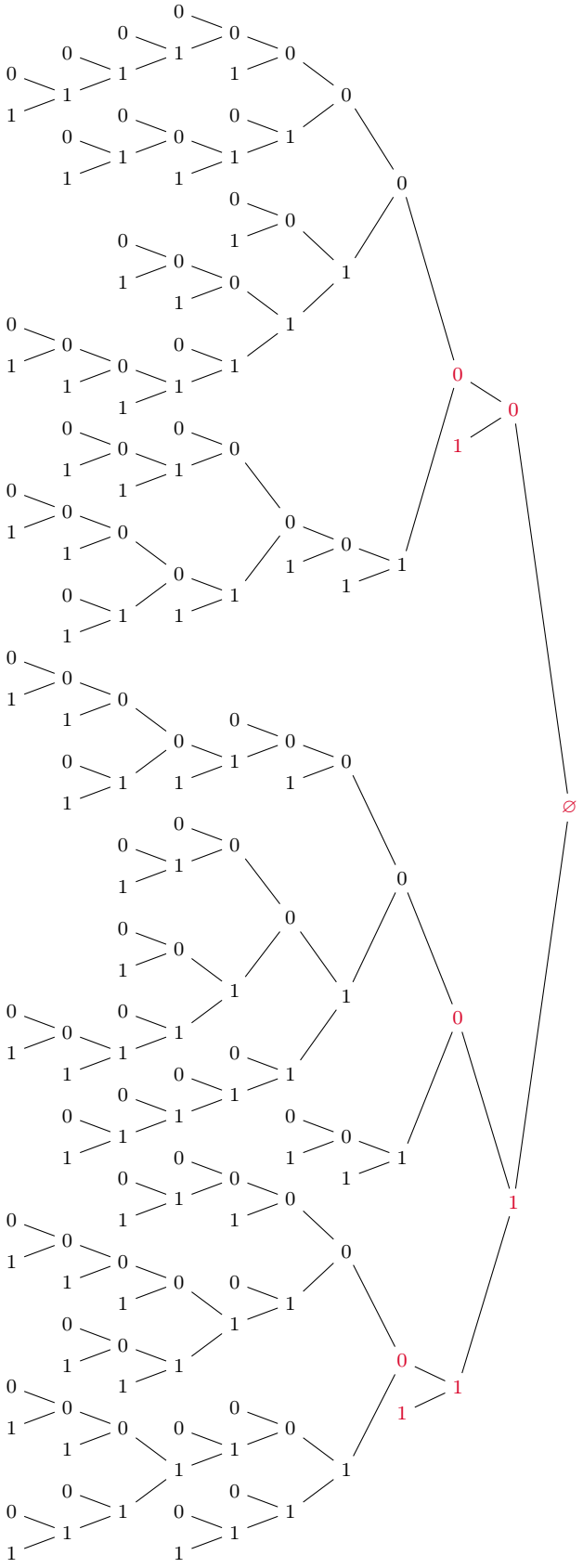

Figura 3: Árvore de contextos de *Cadeia 1*, com esqueleto em vermelho

4.3 Algoritmos

4.3.1 Extração de esqueleto a partir da árvore (algoritmo teórico)

Notação

- Folhas irmãs: conjunto de todas as folhas que diferem apenas no último símbolo (mais longe no passado), com pai χ^*:

$$\chi = \{\chi_j, j = 1, 2, \cdots, |\mathcal{A}| : \chi_j = a_j\chi^*\}.$$

- Ordem: Número de símbolos do passado de um nodo w, representado por l.

Seja τ a árvore probabilística geradora de uma cadeia de Markov de ordem k, com alfabeto $\mathcal{A} = \{a_1, a_2, \cdots, a_{|\mathcal{A}|}\}, h = |\mathcal{A}|$, onde cada folha de ordem $l \leq k$ represente um passado w.

Algoritmo

1. Para cada nodo w de ordem $l \leq k$, associe um vetor de transições possíveis: $t(w) : \mathcal{A}^l \to \{0, 1\}^p$, $t(w) = (t_1, t_2, \cdots, t_p)$, $t_i = \mathbf{1}\{p(a_i \,|\, w) > 0\}$.

2. Para cada conjunto de folhas irmãs χ com pai χ^*, remova χ da árvore τ se $t(\chi_1) = t(\chi_2) = \cdots = t(\chi^*)$. Continue até não haver folhas a serem cortadas.

Obtém-se o *esqueleto* da cadeia.

4.3.2 Estimação de esqueleto a partir de dados (algoritmo empírico)

Notação

- Folhas irmãs: conjunto de todas as folhas que diferem apenas no último símbolo (mais long e no passado), com pai χ^*:

$$\chi = \{\chi_j, j = 1, 2, \cdots, |\mathcal{A}| : \chi_j = a_j\chi^*\}.$$

- Parâmetros de sensibilidade: $\alpha \, e \, \gamma$. Deseja-se detectar probabilidades maiores ou iguais a γ, com nível de significância α.

- Ordem: Número de símbolos do passado de um nodo w, representado por l.

Sejam $Z_1, Z_2, \cdots, Z_n$ observações da CEMAV com alfabeto $\mathcal{A} = \{a_1, a_2, \cdots, a_h\}$, em ordem, $Z_i \in \mathcal{A}, \forall i = 1, \cdots, n$.

Nodos w de ordem l serão construídos com os seguintes atributos:

- passado $s_w \in \mathcal{A}^l$.
- index $\mathcal{L}_w \subseteq \{1, 2, \cdots, n\}$.
- contagem c_w, vetor de tamanho h, $c_w : \mathcal{A} \to \mathbb{N}^h$, $c_w(a_j) = \sum_{i \in \mathcal{L}_w} \mathbf{1}\{Z_i = a_j\}$, contagem do símbolo a_j sobre o índex $\mathcal{L}_w$.

Algoritmo:

1. Defina o número de corte $N_{min} = \lceil \log_{1-\gamma} \alpha - \log_{1-\gamma} h \rceil$.
2. Crie a raiz λ:
   - passado $s_\lambda = \varnothing$.
   - index $\mathcal{L}_\lambda = \{1, 2, \cdots, n\}$.
   - contagem c_λ sobre $\mathcal{L}_\lambda$, $c_\lambda(a_j) = \sum_{i=1}^{n} \mathbf{1}\{Z_i = a_j\}$.
3. Para cada nodo w de ordem l, se

$$\sum_{i=1}^{h} c_w(a_i) \geq N_{min},\ e\ \exists i, j \in [1, h], i \neq j\ t.q.\ c_w(a_i) > 0, c_w(a_j) > 0,$$

   crie os nodos $uw, \forall u \in \mathcal{A}$, de ordem $l + 1$:
   - passado $s_{uw} = us_w$.
   - index $\mathcal{L}_{uw} = \{i \in \mathcal{L}_w : Z_{i-l-1} = u\}$.
   - contagem c_w sobre $\mathcal{L}_{uw}$.

   Até não haver nodos w que satisfaçam as condições.
4. Associe a cada nodo w um vetor de transições possíveis:

$$t(w) : \mathcal{A}^l \to \{0, 1\}^h, t(w) = (t_1, \cdots, t_p), t_i = \mathbf{1}\{c_w(a_i) > 0\}.$$

5. Para cada conjunto de folhas irmãs χ, selecione apenas as que tiverem contagem total maior que N_{min}: $\chi' = \{\chi_j : \sum_{i=1}^{h} c_{\chi_j}(a_i) > N_{min}\}$. Remova todo o conjunto χ da árvore se, e apenas se:

$$t(\chi_j) = t(\chi^*), \forall \chi_j \in \chi',$$

   onde χ^* é o nodo pai de χ. Prossiga até não haver folhas a serem cortadas.

A árvore resultante é a estimação do *esqueleto*.

4.4 Códigos

Um pacote de R foi desenvolvido ao longo da iniciação científica, disponível em pacote skeleton *(ainda em desenvolvimento)*, que será publicado em um artigo posterior. O pacote possui as funcionalidades:

- Estimação de esqueletos a partir de dados, fornecendo a árvore de contextos e a matriz de transição do *esqueleto* estimadas (função *skeleton*).
- Funções separadas para cada passo do item acima:
  - Geração da árvore inicial (*startskel*).
  - Cortes à árvore inicial (*sculptskeleton*).
  - Preenchimento de contextos para obter todos os elementos $s \in \mathcal{A}^d$ (*expand.transitions*).
  - Transformação *Lista de contextos/probabilidades* $\rightarrow$ *Matriz de transições* (*trans.to.matrix*)(operação reversa também, em *matrix.to.trans*).

  Caso o usuário deseje verificar os passos manualmente.
- Análise da irredutibilidade (*irreductible*).
- Simulação de cadeia a partir das transições proibidas (em desenvolvimento).
- Função de lógica que permite conferir se um elemento x está em um intervalo, definindo se os limites do intervalo são abertos ou fechados (*interval.check*).

O repositório da iniciação contém códigos considerados relevantes para o relatório:

- Função *cemav_bin*, permitindo simular uma CEMAV binária que siga uma função $p(1 \mid 10^k) = q_k$, como a apresentada no início da seção 4.2.
- Códigos das simulações das *Cadeias 1 e 2*, resultando nas amostras utilizadas neste projeto.
- Amostras geradas, em formato *txt*.

A diferença na eficiência dos códigos para a *Cadeia 1* e para a *Cadeia 2* (muito mais rápidos) se deve ao ganho de experiência com a linguagem R ao longo do projeto. Os códigos para a simulação da *Cadeia 2* podem ser utilizados (com poucas modificações) para simular a *Cadeia 1,* porém julgou-se interessante manter a "versão antiga", demonstrando o progresso das habilidades de programação do aluno ao longo da iniciação científica.

5 Discussão

5.1 Convergência do algoritmo empírico

Seja $w_s^0 \in \mathcal{S}$ um contexto de esqueleto, sufixo de uma sequência s de tamanho k, e w_s^m sua estimação após m ocorrências da sequência, observada em uma amostra aleatória $z_1, \cdots, z_n$ da cadeia $(Z_n)_{n\in\mathbb{N}}$, que possui $\mathcal{S}$ como esqueleto e $\mathcal{A} = \{a_1, \cdots, a_h\}$ como alfabeto. Continuando, sejam $t_i^0(s) = \mathbf{1}\{p_i^0(s) > 0\},\ p_i^0(s) = p(a_i \,|\, s), a_i \in \mathcal{A}$ elementos do vetor de transições possíveis descrito no algoritmo teórico, e $t_i^m(s)$ sua estimação após m observações de s pelo algoritmo empírico: Sendo $\mathcal{L}_s = \{i : Z_{i-k-1}^{i-1} = s\}$, $|\mathcal{L}_s| = m$, $t_i^m(s) = \mathbf{1}\{[\sum_{j\in\mathcal{L}_s} \mathbf{1}\{z_j = a_i\}] > 0\}$, obtemos o

Lema 2.

$$\lim_{m\to+\infty} \mathbb{P}[t_i^m(s) = t_i^0(s)] = 1.$$

Demonstração. O próximo símbolo da cadeia depende apenas do passado relevante, logo $Z_j \sim Multinom(1, p_1^0(s), \cdots, p_1^h(s)) \Rightarrow \mathbf{1}\{Z_j = a_i\} \sim Ber(p_i^0(s)), \forall j \in \mathcal{L}_s$, e $Y_i^m(s) = \sum_{j\in\mathcal{L}_s} \mathbf{1}\{Z_j = a_i\} \sim Bin(m, p_i^0)$.

$$\mathbb{P}[t_i^m = 0] = \mathbb{P}[Y_i^m(s) = 0] = (1 - p_i^0(s))^m \Rightarrow \lim_{m\to\infty} \mathbb{P}[t_i^m = 0] = \begin{cases} 1,\ p_i^0(s) = 0 \\ 0,\ p_i^0(s) > 0 \end{cases}$$

□

Assumindo que todos os contextos $w_s^0 \in \mathcal{S}$ possuem probabilidade positiva de serem observados, obtemos o

Teorema 1.

$$\lim_{m\to+\infty} \mathbb{P}[w_s^m = w_s^0] = 1.$$

Demonstração. A prova é consequência do lema 2, já que, com observações suficientes, obtemos uma aproximação com a precisão desejada dos vetores de transição teóricos. Portanto, os cortes na árvore estimada incial obtém os mesmos resultados que os do algoritmo teórico. □

5.2 Redução de custo computacional

Abordamos agora a redução de custo resultante do uso do *esqueleto* na análise da irredutibilidade.

Como discutido na seção 4.1.3, uma cadeia de Markov de ordem k e seu *esqueleto* de ordem d, ambos tendo $\mathcal{A}$ como alfabeto, podem ser tratados como cadeias de Markov de ordem 1 com alfabetos $\mathcal{A}^k$ e $\mathcal{A}^d$, respectivamente, e a análise da irredutibilidade de uma cadeia com alfabeto $\mathcal{B}$ envolve até $|\mathcal{B}| - 1$ multiplicações de

matriz. Cada multiplicação de matriz envolve $|\mathcal{B}|^2$ produtos de vetor, resultando em um custo aproximado de $|\mathcal{B}|^3$ produtos de vetor[1].

A extração do *esqueleto* de uma cadeia de ordem k envolve no máximo $|\mathcal{B}|^k$ cortes (número de nodos), no entanto o número real sempre será menor, já que a análise da irredutibilidade seria instantânea em casos que se cortam todos os nodos (nenhuma transição proibida).

Tratando o produto de vetor e a comparação/corte de nodos como operações similares, obtemos os custos aproximados (a aproximação melhora à medida que k *e* d aumentam):

- $|\mathcal{A}|^{3k}$ operações, analisando a irredutibilidade pela cadeia.
- $|\mathcal{A}|^{k+3d}$ operações, analisando a irredutibilidade pelo *esqueleto*.

Portanto, o *esqueleto* ganha em custo em situações que $k > \frac{3d}{2}$, tornando-se mais eficiente na maioria dos casos. Em casos mais discrepantes onde $k >> d$, o termo $|\mathcal{A}|^{3d}$ se torna desprezível e a redução de custo é aproximadamente cúbica. A análise da periodicidade também envolve um máximo de $|\mathcal{B}| - 1$ multiplicações de matrizes, portanto os resultados se aplicam da mesma forma.

Aplicando estes cálculos às *Cadeias 1 e 2*, obtemos as seguintes reduções:

- *Cadeia 1*: $2^{30} \rightarrow 2^{19}$ operações, custo 2048 vezes menor.
- *Cadeia 2*: $4^{15} \rightarrow 4^{14}$ operações, pouca redução, exemplo de cadeia com ordem próxima à do *esqueleto*.

No entanto, em ambos os casos, o Lema 1 se aplica (visível nas Figuras 2 e 3), e podemos declarar que nenhuma das cadeias é irredutível, sem cálculos além dos cortes do *esqueleto*. Nestes casos, apenas os cortes foram efetuados, e a redução se torna $|\mathcal{A}|^{3k} \rightarrow |\mathcal{A}|^k$, independente do tamanho do *esqueleto*.

Isto resulta em um custo $2^{20} = 4^{10} = 1.048.576$ vezes menor, na análise das *Cadeias 1 e 2*.

5.3 Conclusões e Considerações

Foi possível aprofundar consideravelmente meus conhecimentos acerca de cadeias estocásticas ao longo deste projeto, desenvolver habilidades de construção e implementação de algoritmos não exploradas em outros momentos da graduação, e aperfeiçoar a capacidade de programação em R.

O *esqueleto* apresenta grande potencial de ganhos em eficiência em análises envolvendo cadeias estocásticas, sejam CEMAV ou de ordem fixa. A construção e

[1] O custo exato é $|\mathcal{B}|^{3k} - |\mathcal{B}|^{2k}$. A diferença entre o custo exato e o aproximado diminui rapidamente: Para as *cadeias 1 e 2*, é inferior a 0,1%.

implementação de algoritmos para obter o *esqueleto* de uma cadeia são dois passos iniciais importantes na disponibilização dos resultados deste objeto, além de uma grande oportunidade de desenvolvimento pessoal e profissional, e foi possível provar a convergência do algoritmo construido.

A iniciação científica também me proporcionou uma amostra do mundo da pesquisa acadêmica, reforçando o desejo de prosseguir com uma pós-graduação.

6 Material para publicação

O algoritmo teórico obtido será publicado em conjunto com os professores Christophe Gallesco e Daniel Takahashi, em [GOT25].

O algoritmo empírico e o pacote em R serão publicados posteriormente, de forma ainda não definida.

7 Referências

[BL07] Jose Borges e Mark Levene. "Evaluating Variable-Length Markov Chain Models for Analysis of User Web Navigation Sessions". Em: *IEEE Transactions on Knowledge and Data Engineering* 19.4 (2007), pp. 441–452. DOI: 10.1109/TKDE.2007.1012.

[FGG01] Roberto Fernandez, Antonio Galves e Ferrari Galves. *Coupling, renewal and perfect simulation of chains of infinite order*. Jan. de 2001.

[GGT25] Christophe Gallesco, Alexsandro Gallo e Daniel Yasumasa Takahashi. *Uniqueness of stationary compatible probability measure for chains of infinite order with forbidden transitions*. 2025. arXiv: 2507.16981 [math.PR]. URL: https://arxiv.org/abs/2507.16981.

[GOT25] Christophe Gallesco, Caio Théodore G. Huss Oliveira e Daniel Yasumasa Takahashi. "Reduction algorithm for high order Markov chains". Em desenvolvimento. 2025.

[GL08] Antonio Galves e Eva Löcherbach. *Stochastic chains with memory of variable length*. 2008. arXiv: 0804.2050 [math.PR]. URL: https://arxiv.org/abs/0804.2050.

[Gal+12] Antonio Galves et al. "Context tree selection and linguistic rhythm retrieval from written texts". Em: *The Annals of Applied Statistics* (2012). DOI: 10.1214/11-AOAS511.

[Lia+16] Weinan Liao et al. “Alignment-free Transcriptomic and Metatranscriptomic Comparison Using Sequencing Signatures with Variable Length Markov Chains”. Em: *Scientific Reports 6, Nature* (2016). DOI: https://doi.org/10.1038/srep37243.

[Mae24] Martin Maechler. *VLMC: Variable Length Markov Chains ('VLMC') Models*. R package version 1.4-4. 2024. URL: https://CRAN.R-project.org/package=VLMC.

[R24] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2024. URL: https://www.R-project.org/.

[Ris83] Jorma Rissanen. “A universal data compression system”. Em: *Information Theory, IEEE Transactions on* 29 (out. de 1983), pp. 656–664. DOI: 10.1109/TIT.1983.1056741.

[Sau20] Roger M. Sauter. “Advanced R (2nd ed.)” Em: *Technometrics* 62.3 (2020), pp. 417–417. DOI: 10.1080/00401706.2020.1783959. eprint: https://doi.org/10.1080/00401706.2020.1783959. URL: https://doi.org/10.1080/00401706.2020.1783959.

8 Perspectivas de Continuidade

Após a entrega deste relatório, ainda há atividades a serem concluídas:

- Inclusão da função de simulação no pacote *skeleton*.
- Publicação do artigo co-autorado [GOT25].
- Publicação do algoritmo empírico e pacote *skeleton*.
- Análises com o algortimo empírico, que poderão acompanhar a publicação do item anterior.

9 Desempenho acadêmico

A iniciação científica foi feita em conjunto com um estágio, e não houve impacto no desempenho acadêmico:

1. 2º semestre de 2024:
    - ME731 - 8,0
    - ME812 - Aprovado (conceito)

- ME918 - 7,0

2. 1º semestre de 2025:

   - EX011 - 10,0
   - ME623 - 7,7
   - ME705 - 8,6
   - ME821 - Aprovado (disciplina de estágio)
   - ME851 - Aprovado (conceito)
   - ME852 - Aprovado (conceito)

2 tracamentos de matéria foram feitos no segundo semestre de 2024. Não foram por sobrecarga, mas sim para manter a possibilidade de se estender a graduação até o segundo semestre de 2025, o que não foi necessário.

O aluno terminou a graduação em julho, motivo pelo qual a bolsa e o cronograma param neste mês.

10 Apoio e Agradecimentos

O projeto foi financiado por uma bolsa PIBIC/CNPq, com vigência de setembro de 2024 a julho de 2025.

Agradeço ao professor Christophe Frédéric Gallesco pela sua orientação e apoio ao longo do projeto.

Agradeço a Laurent Pascal Huss, meu pai, que me apoiou ao longo de toda a minha vida, e faleceu pouco antes da conclusão deste projeto.