

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE CIÊNCIAS EXATAS E DA TERRA  
DEPARTAMENTO DE ESTATÍSTICA

Félix Lopes Barboza

# **Cadeias Estocásticas com Memória de Alcance Variável**

Natal - RN

Novembro de 2019

Félix Lopes Barboza

## **Cadeias Estocásticas com Memória de Alcance Variável**

Monografia de Graduação apresentada ao Departamento de Estatística do Centro de Ciências Exatas e da Terra da Universidade Federal do Rio Grande do Norte como requisito parcial para a obtenção do grau de Bacharel em Estatística.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE CIÊNCIAS EXATAS E DA TERRA  
DEPARTAMENTO DE ESTATÍSTICA

Orientador: Prof. Dr. Bruno Monte de Castro

Natal - RN  
Novembro de 2019

Universidade Federal do Rio Grande do Norte - UFRN  
Sistema de Bibliotecas - SISBI  
Catalogação de Publicação na Fonte. UFRN - Biblioteca Setorial Prof. Ronaldo Xavier de Arruda - CCET

Barboza, Félix Lopes.

Cadeias estocásticas com memória de alcance variável / Félix  
Lopes Barboza. - 2019.  
50f.: il.

Monografia (Bacharelado) - Universidade Federal do Rio Grande  
do Norte, Centro de Ciências Exatas e da Terra, Departamento de  
Estatística. Natal, 2019.

Orientador: Bruno Monte de Castro.

1. Estatística - Monografia. 2. Algoritmo contexto -  
Monografia. 3. Árvore de contexto - Monografia. 4. BIC -  
Monografia. 5. Cadeias de Markov - Monografia. I. Castro, Bruno  
Monte de. II. Título.

RN/UF/CCET

CDU 519.2

Félix Lopes Barboza

## **Cadeias Estocásticas com Memória de Alcance Variável**

Monografia de Graduação apresentada ao Departamento de Estatística do Centro de Ciências Exatas e da Terra da Universidade Federal do Rio Grande do Norte como requisito parcial para a obtenção do grau de Bacharel em Estatística.

Trabalho aprovado. Natal - RN, 28 de Novembro de 2019 .

Bruno Monte de Castro

Prof. Dr. Bruno Monte de Castro  
Orientador

Antonio Marcos B. do Nascimento

Prof. Dr. Antonio Marcos Batista do  
Nascimento  
Examinador

Francisco Moisés C. de Medeiros

Prof. Dr. Francisco Moisés Cândido  
de Medeiros  
Examinador

Natal - RN  
Novembro de 2019

# Agradecimentos

A Deus pela vida, saúde, força de vontade e perseverança durante toda essa caminhada.

Agradeço aos meus pais Etienne Lopes dos Santos e Abraão Elias Barboza, pelos ensinamentos ao longo da vida e por sempre transmitirem muito amor e carinho.

À minha irmã Eduarda por estar sempre presente, me apoiando nas diversas situações que passei. Minha tia Eliana por desde muito pequeno já me aconselhar bastante aos estudos e sempre refletir persistência e dedicação.

Agradeço ao professor Dr. Bruno Monte de Castro por ter me orientado e auxiliado no desenvolvimento deste trabalho, além de sempre ter sido paciente nas diversas dificuldades que tive ao longo da orientação, sempre se mostrando solícito a me ajudar e transmitir o máximo de conhecimento e conselhos possíveis.

À minha amiga Carla Apolinário por me aguentar por esses quase 4 anos de curso e sempre está presente nas mais diversas situações, me ajudando, apoiando e tornando minha vida muito mais feliz. Aos meus amigos Rayland, Fabrício e Alessandro que sempre estiveram muito presentes ao longo das diversas disciplinas que passamos juntos.

Ao PET, pela ajuda financeira, por proporcionar um vasto conhecimento e interação com alunos do curso de outros períodos, em especial Lucas, Érika, Rodrigo, Felipe, Wesley, Ivonaldo e Thiago. E também aos orientadores do programa Dione e Moisés por sempre passarem inspiração nas atividades desenvolvidas.

Ao departamento de estatística com quem tive uma excelente experiência com os professores e toda estrutura disponível. Ao LEA por proporcionar situações em que um profissional da área de estatística certamente passará.

Agradeço ao TJRN por possibilitar usufruir da experiência de estágio, podendo assim colocar em prática diversos conhecimentos aprendidos durante o curso e com isso trocar diversas experiências com pessoas de outras áreas. Aos diversos amigos que levarei de lá, em especial, Ítalo, Lênora, Elisabeth, Cristiana, Viviane, Iury, Amanda e Dr. Dantas.

Aos meus amigos Francisco Bento, Paula Laís, Matheus Camelo, Vinícius Alves, Mariana Eufasino, Victor Alves, Talita, Isabella e Geovane que mesmo longe do meu ambiente acadêmico foram e são de grande importância para que eu chegasse até aqui e alcançasse esse objetivo.

*“Nós só podemos ver um pouco do futuro,  
mas o suficiente para perceber que há muito a fazer”  
(Alan Turing)*

# Resumo

Cadeias estocásticas com memória de alcance variável, consiste em analisar uma porção do passado, chamado contexto, para prever o próximo símbolo de uma determinada sequência de interesse. Recentemente tem sido usado para modelar dados em diferentes áreas da ciência como biologia, linguística e música. Este trabalho busca definir essa família de cadeia estocástica, apresentar a estimação por máxima verossimilhança das probabilidades de transição dado uma árvore de contextos, apresentar o algoritmo contexto, o método BIC e comparar esses dois métodos no software R.

**Palavras-chave:** Algoritmo Contexto. Árvore de Contexto. BIC. Cadeias de Markov.

# Abstract

Stochastic chains with memory of variable length, consists of analyzing a portion of the past, called context, to predict the next symbol. Recently, they have been used to model up scientific data in areas as different as biology, linguistics and music. This work seeks to define this family of stochastic chains, present the maximum likelihood estimation of transition probabilities given a context tree, present the algorithm context, the BIC method and compare these two methods in software R.

**Keywords:** Algorithm context. BIC. Markov chains. Tree of context.



# Lista de ilustrações

Figura 1 – Ilustração das árvores do Exemplo 3.1.2. . . . .	18
Figura 2 – Árvore de contextos $\tau$ truncada no nível $k = 2$ . . . . .	19
Figura 3 – Exemplo de representação de uma árvore de contextos probabilística. . . . .	20
Figura 4 – Exemplo de ramos de uma árvore. . . . .	23
Figura 5 – Retirando ramo para aplicação do método BIC. . . . .	24
Figura 6 – Árvore de contexto $\tau_1 = \{00, 10, 20, 11, 21, 2\}$ . . . . .	27
Figura 7 – Árvore de contexto $\hat{\tau}_{1_0}$ . . . . .	29
Figura 8 – Árvore de contexto admissível $\hat{\tau}_{1_1}$ . . . . .	30
Figura 9 – Nova Árvore de contexto admissível $\hat{\tau}_{1_2}$ . . . . .	31
Figura 10 – Comparação da árvore $\tau_1$ geradora da amostra com a árvore estimada $\hat{\tau}_{1_n}$ pelo algoritmo de contexto. . . . .	32
Figura 11 – Distribuição das 100 árvores estimadas. . . . .	33
Figura 12 – Distribuição das 100 árvores estimadas com o dobro de símbolos. . . . .	34
Figura 13 – Nova Árvore de contexto admissível $\hat{\tau}_{1_2}$ pelo método BIC. . . . .	36
Figura 14 – Distribuição das 100 árvores estimadas com a metade de símbolos pelo método BIC. . . . .	37

# Lista de tabelas

Tabela 1	– Número de vezes que um suposto contexto aparece na amostra. . . . .	29
Tabela 2	– Valores para calcular $\Delta_n(00)$ . . . . .	30
Tabela 3	– Resultado dos testes de hipótese para as sequências de $u$ , com $\ell(u) = 2$ . . . . .	31
Tabela 4	– Resultado dos testes de hipótese para as sequências $u$ , com $\ell(u) = 1$ . . . . .	32
Tabela 5	– Valores para calcular $\Delta_n(00)$ pelo método BIC. . . . .	35
Tabela 6	– Resultado dos testes de hipótese para as sequências $u$ , com $\ell(u) = 2$ pelo método BIC. . . . .	35
Tabela 7	– Resultado dos testes de hipótese para as sequências $u$ , com $\ell(u) = 1$ pelo método BIC. . . . .	36

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Objetivos</b>	<b>11</b>
<b>2</b>	<b>CADEIAS DE MARKOV</b>	<b>13</b>
<b>2.1</b>	<b>Processos Estocásticos</b>	<b>13</b>
<b>2.2</b>	<b>Propriedade de Markov</b>	<b>14</b>
<b>2.2.1</b>	<b>Inferência em cadeias de Markov</b>	<b>15</b>
<b>3</b>	<b>CADEIAS ESTOCÁSTICAS COM MEMÓRIA DE ALCANCE VA- RIÁVEL</b>	<b>17</b>
<b>3.1</b>	<b>Modelo Probabilístico</b>	<b>17</b>
<b>3.2</b>	<b>Seleção Estatística de Modelos</b>	<b>21</b>
<b>3.2.1</b>	<b>Algoritmo Contexto</b>	<b>23</b>
<b>3.2.2</b>	<b>Método BIC</b>	<b>24</b>
<b>4</b>	<b>SIMULAÇÕES E RESULTADOS</b>	<b>27</b>
<b>4.1</b>	<b>Estimação via algoritmo contexto</b>	<b>28</b>
<b>4.2</b>	<b>Estimação via método BIC</b>	<b>34</b>
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>38</b>
	<b>REFERÊNCIAS</b>	<b>39</b>
	<b>APÊNDICE A – FUNÇÕES CRIADAS NO R</b>	<b>40</b>
	<b>APÊNDICE B – SIMULAÇÕES FEITAS NO R</b>	<b>48</b>

# 1 Introdução

Analisar o passado de uma sequência de símbolos, com o intuito de prever o próximo, é uma tarefa que pode exigir a estimação de muitos parâmetros. Buscar a estrutura fonte que gerou essa sequência com menor complexidade possível é o desejável, pois desse modo estaríamos estimando apenas o que realmente importa na estrutura de dados. Imaginemos um sistema composto por uma lâmpada, um observador e uma porta entre ambos. A lâmpada possui dois estados possíveis: acesa, representada por 1, ou apagada, representada por 0. Quando a lâmpada está acesa o observador pode receber a luz emitida através da porta, que também pode se encontrar em dois estados: aberta, representada por 1, ou fechada, representada por 0. Estes estados independem um do outro. Se definirmos uma sequência  $(Z_n)_{n \in \mathbb{N}}$  que representa se o observador conseguiu ou não receber a luz, utilizando uma árvore de contextos é possível representar os estados passados da sequência, mostrando qual é o relevante para identificar o próximo estado. A cadeia estocástica  $(Z_n)_{n \in \mathbb{N}}$  é, então, uma cadeia com memória de alcance variável.

A classe das cadeias estocásticas com memória de alcance variável foi introduzida em 1983 por Jorma Rissanen no artigo *A universal data compression system* (RISSANEN, 1983). Essa classe ficou popularizada na comunidade estatística e probabilística por P. Bühlmann e A. J. Wyner, em 1999, no artigo *Variable Length Markov Chains*. (em português, Cadeias de Markov de Alcance Variável) (BÜHLMANN; WYNER et al., 1999). A designação Cadeias Estocásticas com Memória de Alcance Variável, se deve a Galves e Löcherbach, em 2008, no artigo *Stochastic chains with memory of variable length* (em português, Cadeias estocásticas com memória de alcance variável) (GALVES; LÖCHERBACH, 2008).

## 1.1 Objetivos

Este trabalho tem o interesse de analisar o comportamento de fenômenos que utilizam uma amostra de elementos simbólicos, com objetivo de prever o próximo símbolo desse fenômeno. Mais precisamente, estamos interessados em descrever a estrutura fonte que gerou essa sequência com menor complexidade possível. Por exemplo, quando estamos estudando linguística, mais especificamente em gramática, existe uma noção que para prever as próximas letras de uma palavra precisamos olhar uma porção do passado de forma que a sentença faça sentido (GALVES et al., 2012). Essa mesma ideia pode ser usada para prever sequências de notas musicais.

O modelo mais simples para prever o próximo símbolo é aquele em que necessitamos olhar somente para o que acontece no passado mais recente e é conhecido como

Cadeias de Markov. Existem situações em que o passado mais recente não é suficiente para prever o próximo símbolo, há casos em que precisamos observar  $k$  passos anteriores, assim definimos uma generalização do modelo como cadeia de Markov de ordem  $k$ . A maior desvantagem das cadeias de Markov de ordem  $k$  é que em alguns casos estamos estimando mais probabilidades do que o necessário. Ou seja, dependendo do passado, basta olhar uma porção variável do mesmo para obter o próximo passo. Esses modelos são conhecidos como cadeias estocásticas com memória de alcance variável. Aplicação desse modelo pode ser encontrados em biologia molecular (BUSCH et al., 2009), linguística (GALVES et al., 2012), entre outras áreas.

Podemos resumir os objetivos específicos deste trabalho como;

- Apresentar definições e propriedades das cadeias de Markov;
- Calcular o estimador de máxima verossimilhança das probabilidades de transição;
- Apresentar conceitos das cadeias estocásticas com memória de alcance variável utilizando exemplos e ilustrações;
- Estudar critérios de seleção da árvore de contextos;
- Criar algoritmos que possam reproduzir os critérios estudados;
- Discutir o desempenho dos estimadores de cada critério.

Para realização das simulações será utilizado o software R (R Core Team, 2019).

## 2 Cadeias de Markov

As cadeias de Markov foram introduzidas pelo matemático russo Andrei Andreyevich Markov, em torno de 1905. Sua intenção era criar um modelo probabilístico para analisar a aparição de vogais em poemas e textos literários.

Essas cadeias possuem a seguinte característica, pode-se fazer previsões do futuro com base somente do seu estado atual, independentemente do que aconteceu no passado. Em outras palavras, condicionando ao estado atual da cadeia, suas evoluções futuras e passadas são independentes. As cadeias de Markov são bastantes gerais e desfrutam de uma ampla gama de aplicações. Por exemplo, na física, são amplamente utilizadas na mecânica estatística. Nas ciências da informação (PRATAS et al., 2015), elas são usadas no processamento de sinais, codificação, compactação de dados e reconhecimento de padrões. Em finanças (HAMILTON, 1989), elas são usadas para descrever a evolução dos preços dos ativos. Elas também são usadas em jogos, músicas, genética, beisebol, história e assim por diante.

### 2.1 Processos Estocásticos

Dentro da teoria das probabilidades, um processos estocástico é uma sequência de variáveis aleatórias representando a evolução de fenômenos (ou sistemas) que são observados ao longo do tempo. Assim, ao invés de descrevermos o sistema através de equações determinísticas, que dado uma condição inicial, conhecemos toda a evolução do sistema, vamos utilizar processos estocásticos, para o qual, dado uma condição inicial, ainda temos diversas trajetórias possíveis para a evolução do fenômenos (ou sistemas).

**Definição 2.1.1.** *Dado um espaço de probabilidade  $(\Omega, \mathcal{F}, \mathbb{P})$ , um processo estocástico é uma sequência de variáveis aleatórias  $\{X_{(t)} : t \in T\}$  parametrizado por um conjunto  $T$ , chamado de espaço paramétrico, e assumindo valores em um conjunto  $A$ , chamado espaço de estado (ou alfabeto).*

Se  $A$  é um espaço de estado, o evento  $\{X_{(t)} \in A\}$  corresponde a situação em que em um tempo  $t$ , o processo assume algum valor dentro do conjunto  $A$ .

Os diferentes tipos de processos estocásticos se obtém ao considerar as diferentes possibilidades para: o espaço paramétrico, o espaço de estado, as características das trajetórias, e principalmente a relação de dependência entre as variáveis aleatórias que compõem o processo.

No estudo em questão estaremos interessados em um processo estocástico específico, uma cadeia de Markov, com espaço paramétrico sendo o conjunto dos números inteiros,  $T = \mathbb{Z}$ , e espaço de estados sendo  $A$  com um número enumerável de símbolos. Esse processo consiste apenas na dependência do estado imediatamente anterior, para predição do próximo símbolo, ignorando assim todo o passado que o precedeu.

## 2.2 Propriedade de Markov

A cadeia de Markov é um processo estocástico com a propriedade de Markov. Para simplificar a notação, vamos considerar que  $\mathbb{P}(X_n = x_n)$  como  $p(x_n)$ . E o significado da probabilidade condicional  $p(x_{n+1}|x_n)$  é análogo.

**Definição 2.2.1.** *Uma cadeia de Markov é um processo estocástico a tempo discreto  $(X_n)_{n \in \mathbb{Z}}$ , com espaço de estados discreto, e que satisfaz a propriedade de Markov, isto é, para qualquer inteiro  $n \geq 0$ , e para quaisquer estados  $x_0, \dots, x_{n+1}$ , se cumpre*

$$p(x_{n+1}|x_0, \dots, x_n) = p(x_{n+1}|x_n) \quad (2.1)$$

Se considerarmos o tempo  $n + 1$  como um tempo futuro, o tempo  $n$  como o presente e os tempos  $0, 1, \dots, n - 1$  como o passado, então a equação 2.1 estabelece que a distribuição de probabilidade do estado do processo no tempo futuro ( $n + 1$ ) depende unicamente do estado do processo no tempo  $n$ , e não depende dos estados nos tempos passados  $0, 1, \dots, n - 1$ .

As probabilidades  $\mathbb{P}(X_{n+1} = j | X_n = i)$ , que podemos denotar por  $p_{ij}(n, n + 1)$ , representam as probabilidades de transição do estado  $i$  no tempo  $n$ , ao estado  $j$  no tempo  $n + 1$ . Estas probabilidades são conhecidas por **probabilidades de transição** em um passo. Quando os números  $p_{ij}(n, n + 1)$  não dependem de  $n$  dizemos que a cadeia é estacionária e homogênea no tempo. Neste estudo vamos considerar esse caso, de modo que denotaremos as probabilidades de transição em um passo por  $p(j|i) = p_{ij}(n, n + 1)$ . Variando os índices  $i$  e  $j$ , sobre o conjunto de estados  $\{0, 1, 2, \dots\}$ , obtemos a matriz de probabilidade de transição em um passo, que vamos denotá-la por  $P$ .

$$P = \begin{pmatrix} p(0|0) & p(1|0) & p(2|0) & \cdots \\ p(0|1) & p(1|1) & p(2|1) & \cdots \\ p(0|2) & p(1|2) & p(2|2) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

**Proposição 2.2.1.** *A matriz de probabilidade de transição  $P = (p(j|i))_{i,j \in A^2}$  cumpre as seguintes propriedades.*

a)  $p(j|i) \geq 0 \quad \forall i, j \in A$

$$b) \sum_j p(j|i) = 1 \quad \forall i \in A$$

*Demonstração:*

A primeira condição é evidente pelo fato de que os números  $p(j|i)$  são probabilidades. Para a segunda propriedade temos que para qualquer estado  $i$  e qualquer inteiro  $n \geq 0$ ,

$$\begin{aligned} 1 &= \mathbb{P}(\{X_{n+1} = 0\} \cup \{X_{n+1} = 1\} \cup \{X_{n+1} = 2\} \cup \dots) \\ &= \mathbb{P}(\{X_{n+1} = 0\} \cup \{X_{n+1} = 1\} \cup \{X_{n+1} = 2\} \cup \dots | X_n = i) \\ &= \mathbb{P}\left(\bigcup_j X_{n+1} = j | X_n = i\right) \\ &= \sum_j \mathbb{P}(X_{n+1} = j | X_n = i) \\ &= \sum_j p(j|i) \end{aligned}$$

■

### 2.2.1 Inferência em cadeias de Markov

Muitas vezes estamos diante de um conjunto de dados gerado de uma cadeia de Markov conhecida, porém desconhecemos sua matriz de probabilidade de transição. Uma maneira de estimar essa matriz é utilizando a teoria de máxima verossimilhança.

Dado uma amostra  $\{X_0^n = a_0^n\} = \{X_0 = a_0, X_1 = a_1, \dots, X_n = a_n\}$  gerada por uma cadeia de Markov  $(X_n)_{n \geq 0}$  assumindo valores no alfabeto  $A$ , com matriz de probabilidade de transição  $P = \{p(j|i) : j, i \in A\}$ . Estamos interessados em calcular a função de verossimilhança da matriz  $P$ . Por definição,

$$\begin{aligned} L(P|X_0^n = a_0^n) &= \mathbb{P}(X_0 = a_0, X_1 = a_1, \dots, X_n = a_n) \\ &= \mathbb{P}(X_0 = a_0) \mathbb{P}(X_1 = a_1 | X_0 = a_0) \cdot \dots \cdot \mathbb{P}(X_n = a_n | X_{n-1} = a_{n-1}) \\ &= \mathbb{P}(X_0 = a_0) \prod_{m=1}^n \mathbb{P}(X_m = a_m | X_{m-1} = a_{m-1}) \end{aligned}$$

Para continuarmos o desenvolvimento da função de verossimilhança, vamos definir uma função  $N : A^2 \rightarrow \mathbb{N}$ , que representa o número de vezes que observamos o símbolo  $i$  seguido pelo símbolo  $j$  na amostra.

$$N_n(i, j) = \sum_{m=1}^n \mathbb{I}_{\{X_{m-1}=i, X_m=j\}} \quad (2.2)$$

Sendo  $\mathbb{I}$  a função indicadora.

Com essa função definida, podemos escrever a função de verossimilhança, como

$$L(P|X_0^n = a_0^n) = \mathbb{P}(X_0 = a_0) \prod_{j \in A} \prod_{i \in A} p(j|i)^{N_n(i,j)} \quad (2.3)$$



Denotemos por  $l(P|X_0^n = a_0^n)$  a função log da verossimilhança,

$$l(P|X_0^n = a_0^n) = \log \mathbb{P}(X_0 = a_0) + \sum_{j \in A} \sum_{i \in A} N_n(i, j) \log p(j|i)$$

Pela Proposição 2.2.1 sabemos que  $\sum_{j \in A} p(j|i) = 1$ . Com isso, vamos utilizar o método dos multiplicadores de Lagrange. Sejam  $\lambda \in \mathbb{R}$  e  $F(\lambda, P)$  definido por

$$F(\lambda, P) = \log \mathbb{P}(X_0 = a_0) + \sum_{j \in A} \sum_{i \in A} \{N_n(i, j) \log p(j|i) + \lambda[1 - \sum_{i \in A} p(j|i)]\}$$

Derivando  $F(\lambda, p)$  em relação à  $\lambda$  e igualando a zero, temos que

$$\frac{\partial}{\partial \lambda} F(\lambda, p) = 1 - \sum_{i \in A} p(j|i)$$

$$1 - \sum_{j \in A} p(j|i) = 0. \quad (2.4)$$

Derivando  $F(\lambda, p)$  em relação à  $p(j|i)$  e igualando a zero, temos que

$$\frac{\partial}{\partial p(j|i)} F(\lambda, p) = N_n(i, j) \frac{1}{p(j|i)} - \lambda$$

$$\hat{p}(j|i) = \frac{N_n(i, j)}{\lambda}, \quad (2.5)$$

Pelas equações (2.4) e (2.5), temos que

$$\sum_{l \in A} \frac{N_n(i, l)}{\lambda} = 1 \Rightarrow \lambda = \sum_{l \in A} N_n(i, l).$$

Logo, o estimador de máxima verossimilhança é dado por

$$\hat{p}(j|i) = \frac{N_n(i, j)}{\sum_{l \in A} N_n(i, l)}. \quad (2.6)$$

## 3 Cadeias Estocásticas com Memória de Alcance Variável

As cadeias com memória de alcance variável foram introduzidas por Rissanen (1983), para referir-se a cadeias de Markov de ordem finita, onde dependendo do passado, basta olhar uma porção variável dele para obter o próximo passo.

### 3.1 Modelo Probabilístico

Seja  $A$  um alfabeto finito. Para quaisquer  $m, n \in \mathbb{Z}$  com  $m \leq n$ , denotamos por  $s_m^n = (s_m, s_{m+1}, \dots, s_n)$  uma sequência de símbolos em  $A$ . O comprimento de uma sequência será denotada por  $\ell(\cdot)$ , de modo que  $\ell(s_m^n) = n - m + 1$ . A sequência vazia é representada por  $\emptyset$  e seu comprimento é  $\ell(\emptyset) = 0$ .

Se  $u$  e  $v$  são sequências de elementos de  $A$ , denotamos por  $uv$  uma sequência obtida da concatenação de  $u$  e  $v$ . Em particular  $\emptyset u = u\emptyset = u$ .

Dizemos que uma sequência  $v$  é sufixo de uma sequência  $u$  se existe uma sequência  $w$ , com  $\ell(w) \geq 1$ , tal que  $u = wv$  e denotamos esse fato por  $v \preceq u$ . Quando  $u \neq v$  dizemos que  $v$  é um sufixo próprio de  $u$  e denotamos por  $v \prec u$ .

**Exemplo 3.1.1.** Considere o alfabeto  $A = \{\clubsuit, \spadesuit, \diamondsuit, \heartsuit\}$  e seja  $u = \clubsuit\diamondsuit\heartsuit\heartsuit\clubsuit$  e  $v = \heartsuit\heartsuit\clubsuit$  duas sequências de símbolos de  $A$ . A sequência  $v$  é sufixo próprio de  $u$ , pois  $u = \clubsuit\diamondsuit v$ . A concatenação de  $u$  e  $v$  é  $uv = \clubsuit\diamondsuit\heartsuit\heartsuit\clubsuit\heartsuit\heartsuit\clubsuit$ .

Seja  $A^k$  o conjunto de todas sequências sob  $A$  de tamanho  $k$ , vamos denotar por  $A^* := \bigcup_{k=1}^{\infty} A^k$ .

**Definição 3.1.1.** Um subconjunto finito  $\tau$  de sequência de elementos de  $A^*$  é uma árvore de contextos se satisfaz as seguintes condições:

- i) *Propriedade do sufixo:* Para nenhum  $w \in \tau$ , temos  $u \in \tau$  com  $u \prec w$ .
- ii) *Irreduzibilidade:* Nenhuma sequência em  $\tau$  pode ser substituída por um sufixo próprio sem violar a propriedade de sufixo, isto é, dado  $w \in \tau$ , se substituirmos  $w$  por  $s$ , em que  $s \prec w$ , então existe um  $v \in \tau$  tal que  $s \prec v$ , violando a propriedade de sufixo.

Cada elemento de  $\tau$  é chamado contexto. Podemos identificar  $\tau$  como sendo o conjunto de folhas de uma árvore com um número finito de ramos.

**Exemplo 3.1.2.** Vamos verificar se os conjuntos  $\tau$  abaixo são árvores de contextos ou não. Vamos considerar, para esse exemplo o alfabeto  $A = \{0, 1\}$ .

- a) Seja  $\tau = \{0, 1, 01\}$ . Neste caso a propriedade de sufixo não é satisfeita, pois, com efeito, se 01 pertence a  $\tau$  então 1 não pode pertencer a  $\tau$ , pois  $1 \prec 01$ . Portanto, esse conjunto  $\tau$  não é uma árvore de contextos.
- b) Seja  $\tau = \{000, 10, 01, 11\}$ . Neste caso a propriedade de irredutibilidade não é satisfeita, isto é, podemos substituir a sequência 000 por 00, que é um sufixo próprio dela, de tal forma que não existe uma sequência  $v$  no conjunto  $\tau$  que satisfaça  $00 \prec v$ . Em outras palavras, em cada ramo da árvore deve existir no mínimo duas folhas.
- c) Seja  $\tau = \{00, 10, 1\}$ . Este conjunto é uma árvore de contextos, pois satisfaz a propriedade do sufixo e a regra de irredutibilidade.

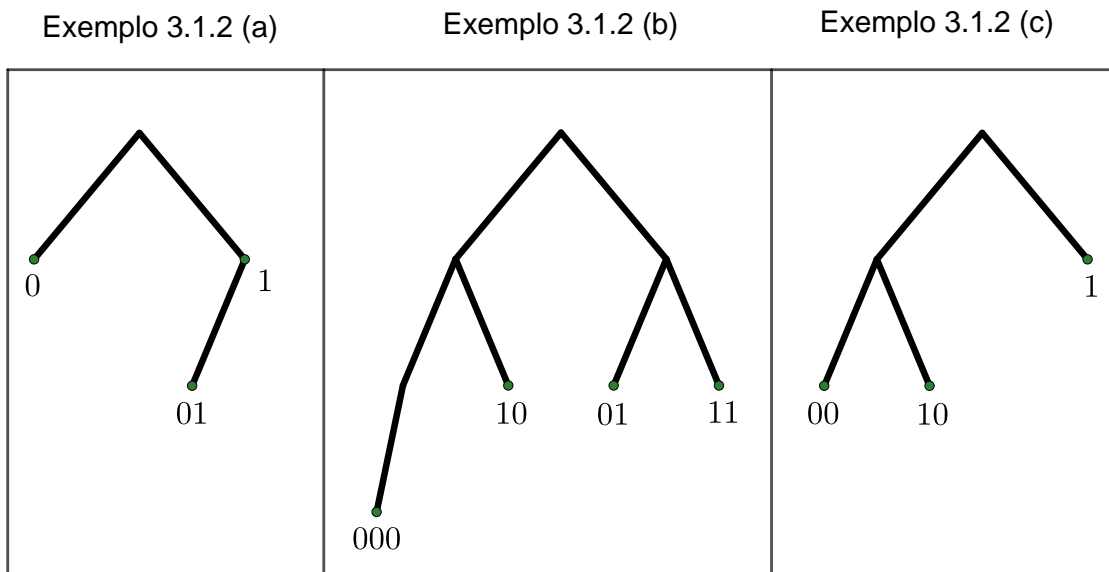


Figura 1 – Ilustração das árvores do Exemplo 3.1.2.

**Definição 3.1.2.** A profundidade de uma árvore de contexto  $\tau$  é definida por

$$\mathcal{L}(\tau) = \max\{\ell(w) : w \in \tau\}$$

**Definição 3.1.3.** Dado uma árvore de contextos  $\tau$  e um número natural  $k$ ,  $\tau|_k$  representa a árvore truncada no nível  $k$ , ou seja,

$$\tau|_k = \{w \in \tau : \ell(w) \leq k\} \cup \{w \in A^k : w \prec u \text{ para algum } u \in \tau, \ell(u) > k\}$$

A Figura 2 a seguir apresenta uma árvore de contextos  $\tau$ , com alfabeto de comprimento 3 e  $\mathcal{L}(\tau) = 3$ . Também é apresentado a árvore truncada no nível  $k = 2$ .

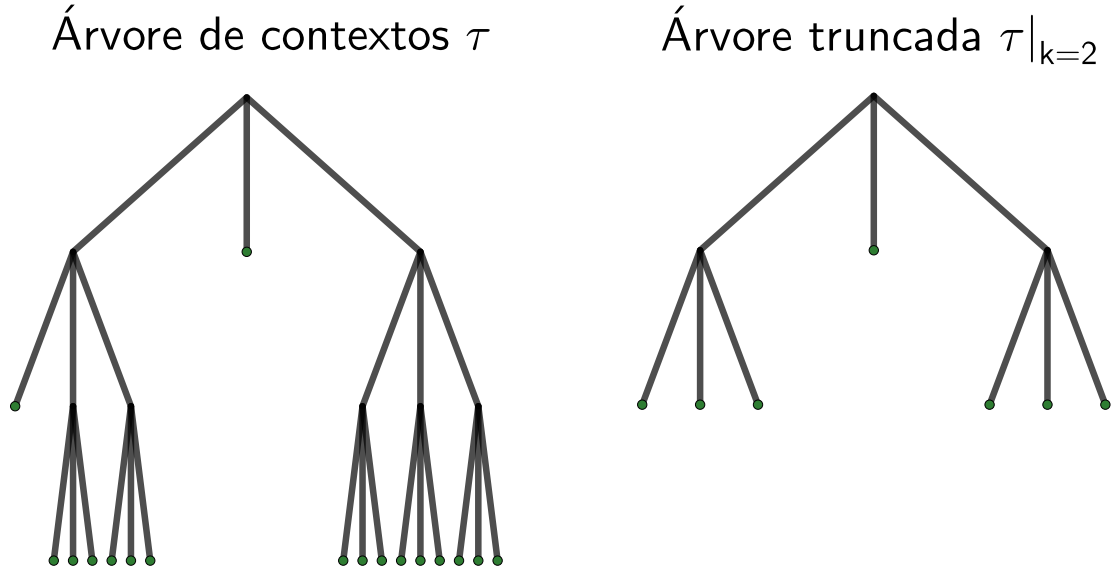


Figura 2 – Árvore de contextos  $\tau$  truncada no nível  $k = 2$ .

**Definição 3.1.4.** Uma árvore de contexto é dita probabilística em  $A$ , se temos um par ordenado  $(\tau, p)$ , tal que

- i)  $\tau$  é uma árvore de contexto;
- ii)  $p = \{p(\cdot|w) : w \in \tau\}$  é uma família de probabilidades em  $A$ , indexada por elementos de  $\tau$ .

Em geral  $p$  é representado por uma matriz, em que cada linha denota um contexto  $w$  e cada coluna um símbolo  $a \in A$  da árvore  $\tau$ . Na matriz de transição abaixo temos um caso em que  $|A| = n$  e  $|\tau| = m$ .

$$P = \begin{matrix} & a_1 & a_2 & \dots & a_n \\ \begin{matrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{matrix} & \begin{pmatrix} \mathbb{P}(a_1|w_1) & \mathbb{P}(a_2|w_1) & \dots & \mathbb{P}(a_n|w_1) \\ \mathbb{P}(a_1|w_2) & \mathbb{P}(a_2|w_2) & \dots & \mathbb{P}(a_n|w_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{P}(a_1|w_m) & \mathbb{P}(a_2|w_m) & \dots & \mathbb{P}(a_n|w_m) \end{pmatrix} \end{matrix}$$

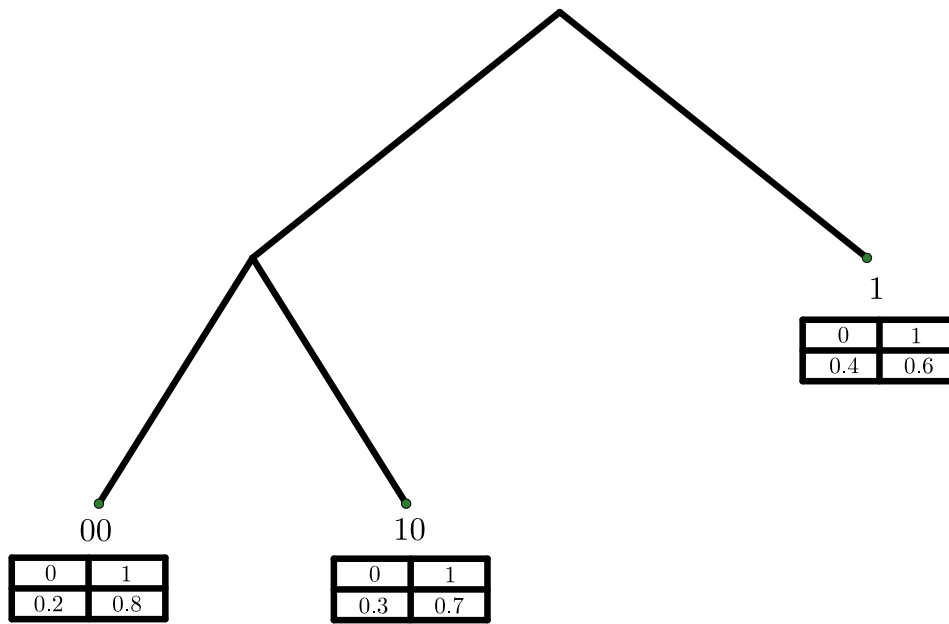


Figura 3 – Exemplo de representação de uma árvore de contextos probabilística.

**Definição 3.1.5.** *Uma sequência  $w$  de elementos de  $A$  é um contexto do processo  $(X_t)_{t \in \mathbb{Z}}$  se  $p(w) > 0$  e se para toda sequência  $u$  de elementos de  $A$  tal que  $w \prec u$ , temos que*

$$\mathbb{P}(X_0 = a | X_{-\ell(u)}^{-1} = u) = p(a|w), \text{ para todo } a \in A$$

É importante destacar que nas probabilidades de transição, o passado é indicado do símbolo mais recente ao símbolo mais remoto:

$$p(b|a_{-1}a_{-2}\dots a_{-k}) = p(b|a_{-k}^{-1}) = \mathbb{P}(X_0 = b | X_{-1} = a_{-1}, \dots, X_{-k} = a_{-k})$$

Por exemplo, se temos uma árvore  $\tau = \{00, 10, 1\}$ , a mesma do Exemplo 3.1.2 (c). Na Figura 3 inserimos as probabilidades de transição de forma que o Exemplo 3.1.2 agora represente uma árvore probabilística de contextos. Uma maneira de representarmos a matriz de probabilidades de transição é

$$P = \begin{matrix} & \begin{matrix} 0 & 1 \end{matrix} \\ \begin{matrix} 00 \\ 10 \\ 1 \end{matrix} & \begin{pmatrix} \mathbb{P}(0|00) & \mathbb{P}(1|00) \\ \mathbb{P}(0|01) & \mathbb{P}(1|01) \\ \mathbb{P}(0|1) & \mathbb{P}(1|1) \end{pmatrix} \end{matrix}$$

O contexto 10, dentro da probabilidade condicional fica 01, isto é, o símbolo mais recente é representado primeiro.

**Definição 3.1.6.** *Seja  $(\tau, p)$  uma árvore probabilística de contextos em  $A$ . Dizemos que o processo estocástico  $(X_t)_{t \in \mathbb{Z}}$  é uma cadeia de alcance variável compatível com  $(\tau, p)$  se*

*i)  $w \in \tau \Leftrightarrow w$  é um contexto para  $(X_t)_{t \in \mathbb{Z}}$ ;*

*ii) para cada  $w \in \tau$ ,  $p(a|w) = \mathbb{P}(X_0 = a | X_{-\ell(w)}^{-1} = w)$ , para todo  $a \in A$ .*

## 3.2 Seleção Estatística de Modelos

A seleção de um modelo, do ponto de vista estatístico, busca-se o modelo mais parcimonioso, isto é, o modelo que envolva o mínimo de parâmetros possíveis a serem estimados e que explique bem o comportamento da variável de interesse. Nesta seção, vamos apresentar algumas abordagens para estimar, sob um critério, a árvore de contextos com base em uma amostra de símbolos.

Considere uma amostra  $X_1^n$  de símbolos em um alfabeto finito  $A$ . Inicialmente, vamos considerar que a amostra foi gerada a partir de uma cadeia com memória de alcance variável  $(X_n)_{n \in \mathbb{Z}}$  e que conhecemos a árvore de contextos  $\tau$ . Queremos encontrar as estimativas das probabilidades de transição  $p(a|u)$ ,  $\forall a \in A$  e  $\forall u \in \tau$ .

Dada uma amostra  $X_1^n$  de símbolos em  $A$  e uma sequência finita  $u \in A^*$ , denotamos por  $N_n(u)$  o número de ocorrências da sequência  $u$  na amostra  $X_1^n$ , e é da forma

$$N_n(u) = \sum_{t=\ell(u)}^n \mathbb{I}_{\{X_{t-\ell(u)+1}^t = u\}}.$$

**Teorema 3.2.1.** *Suponha que  $X_1^n$  é uma amostra de uma cadeia estocástica com memória de alcance variável seguindo uma árvore de contextos  $(\tau, p)$ . O estimador de máxima verossimilhança para  $p(a|u) = \mathbb{P}(X_0 = a | X_{-\ell(u)}^{-1} = u)$  é*

$$\hat{p}_n(a|u) = \begin{cases} \frac{N_n(ua)}{\sum_{a' \in A} N_n(ua')} & , \text{ se } \sum_{a' \in A} N_n(ua') > 0 \\ 0 & , \text{ se } \sum_{a' \in A} N_n(ua') = 0 \end{cases}$$

*Demonstração:*

Seja  $X_n$  uma cadeia estocástica com memória de alcance variável, assumindo valores no alfabeto  $A$ , seguindo uma árvore de contexto  $(\tau, p)$ , com probabilidades de transição  $p = \{p(a|u) : a \in A, u \in \tau\}$ .

Como conhecemos a árvore de contextos, considere que  $v$  é o contexto inicial na amostra. De 2.3 a função de verossimilhança com base na amostra  $X_{-\ell(v)}^n$  é

$$\begin{aligned} L_\tau(X_{-\ell(v)}^n) &= \mathbb{P}(X_{-\ell(v)} = a_{-\ell(v)}, \dots, X_n = a_n) \\ &= \mathbb{P}(X_{-\ell(v)} = a_{-\ell(v)}) \prod_{u \in \tau} \prod_{a \in A} p(a|u)^{N(ua)} \end{aligned}$$

Disso, segue analogamente do que foi visto na seção de cadeia de Markov o resultado

$$\hat{p}_n(a|u) = \begin{cases} \frac{N_n(ua)}{\sum_{a' \in A} N_n(ua')} & , \text{ se } \sum_{a' \in A} N_n(ua') > 0 \\ 0 & , \text{ se } \sum_{a' \in A} N_n(ua') = 0 \end{cases}$$

Considere que a árvore probabilística de contextos  $\tau$  é conhecida, mas a família de probabilidade de transição  $p_n(a|u)$  é desconhecida. Estamos interessados em realizar inferência sobre  $p_n(a|u)$ .

**Exemplo 3.2.1.** *Seja  $X_n$  uma cadeia estocástica com memória de alcance variável sob o alfabeto  $A = \{0, 1\}$  e árvore probabilística de contextos  $\tau = \{00, 10, 1\}$ . E seja a amostra  $X_1^{20} = (1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1)$ .*

*Estamos interessados em estimar as probabilidades de transição:  $\hat{p}(0|00)$ ,  $\hat{p}(1|00)$ ,  $\hat{p}(0|01)$ ,  $\hat{p}(1|01)$ ,  $\hat{p}(0|1)$ ,  $\hat{p}(1|1)$ . Para isso devemos calcular  $N_{20}(000)$ ,  $N_{20}(001)$ ,  $N_{20}(100)$ ,  $N_{20}(101)$ ,  $N_{20}(10)$ ,  $N_{20}(11)$ . Observando a amostra dada, obtemos:*

$$\boxed{N_{20}(000)=1} \quad \boxed{N_{20}(001)=5} \quad \boxed{N_{20}(100)=5} \quad \boxed{N_{20}(101)=0} \quad \boxed{N_{20}(10)=5} \quad \boxed{N_{20}(11)=3}$$

*Com esses valores conseguimos estimar as probabilidades de transição:*

$$\begin{aligned} \hat{p}(0|00) &= \frac{1}{1+5} = \frac{1}{6} \approx 0.17, & \hat{p}(1|00) &= \frac{5}{1+5} = \frac{5}{6} \approx 0.83, \\ \hat{p}(0|01) &= \frac{5}{5+0} = \frac{5}{5} = 1, & \hat{p}(1|01) &= \frac{0}{5+0} = \frac{0}{5} = 0, \\ \hat{p}(0|1) &= \frac{5}{5+3} = \frac{5}{8} = 0.625, & \hat{p}(1|1) &= \frac{3}{5+3} = \frac{3}{8} = 0.375. \end{aligned}$$

Com isso, encontramos a matriz de probabilidade de transição,

$$P = \begin{matrix} & & 0 & 1 \\ \begin{matrix} 00 \\ 10 \\ 1 \end{matrix} & \begin{pmatrix} 0.17 & 0.83 \\ 1 & 0 \\ 0.625 & 0.375 \end{pmatrix} \end{matrix}$$

Geralmente não conhecemos o modelo associado a amostra, ou seja, a árvore de contextos é desconhecida. Precisamos achar mecanismos para estimar a árvore de contexto do modelo e conseqüentemente estimar as probabilidades de transições. Nas subseções seguintes vamos tratar desses mecanismos.

**Definição 3.2.1.** *Seja  $\tau$  uma árvore de contextos e  $u$  uma sequência finita de símbolos em  $A$ , definimos um ramo em  $\tau$  induzido por  $u$  como o conjunto  $B_\tau(u) = \{w \in \tau : w = au \text{ para algum } a \in A\}$*

Na Figura 4 representamos pelo quadro azul o ramo induzido pela sequência  $u = 0$ , ou seja, o ramo  $B_\tau(0)$  tem os filhos  $\{00, 10, 20\}$  enquanto isso no quadro vermelho denotamos o ramo  $B_\tau(1)$ , que possui apenas 2 filhos que são  $\{11, 21\}$ .

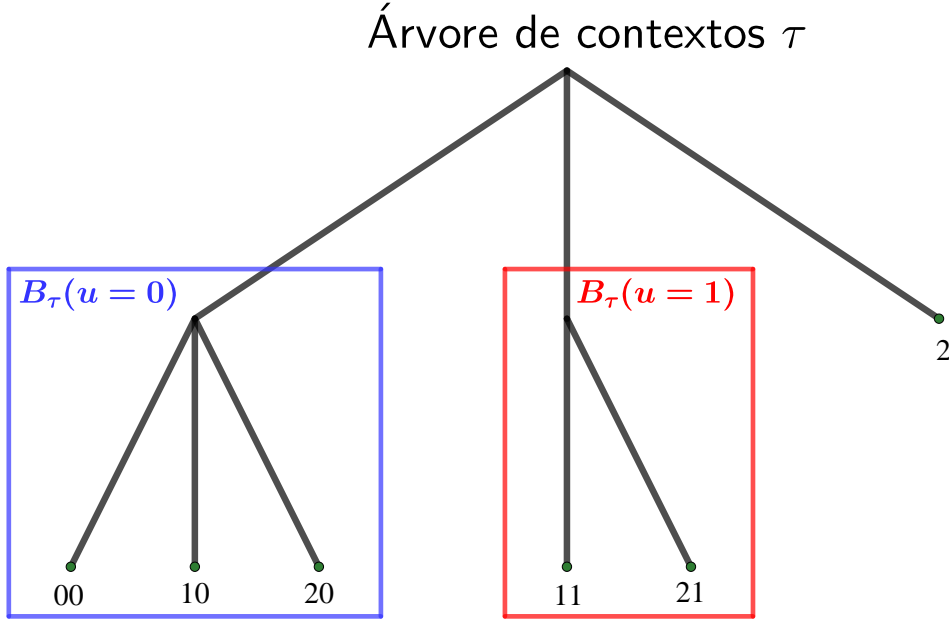


Figura 4 – Exemplo de ramos de uma árvore.

**Definição 3.2.2.** *Seja  $K$  um número inteiro tal que  $1 \leq K < n$ . Uma árvore de contexto admissível de tamanho  $K$ , para a amostra  $X_1^n$ , é qualquer árvore  $\tau$  satisfazendo*

- i)  $u \in \tau$ , se e somente se,  $\ell(u) \leq K$  e  $N_n(u) \geq 1$ .*
- ii) Qualquer sequência  $w \in A^*$  com  $N_n(w) \geq 1$  é sufixo de algum  $u \in \tau$  ou tem um sufixo  $u \in \tau$ .*

Denotamos por  $\mathcal{T}_n$  a maior árvore de contexto admissível.

### 3.2.1 Algoritmo Contexto

Vamos agora descrever um critério de seleção da árvore de contextos estimada proposto por (RISSANEN, 1983). Vamos iniciar com a maior árvore admissível  $\mathcal{T}_n$  de altura  $K$ , sendo  $K$  um número inteiro tal que  $1 \leq K < n$ . Para qualquer sequência  $u \in A^*$  de comprimento máximo e tal que  $|B_{\mathcal{T}_n}(u)| \geq 2$ , queremos testar a hipótese nula

$$H_0^{(u)} : p(a|w) = p(a|u), \quad \forall a \in A, \quad \forall w \in B_{\mathcal{T}_n}(u).$$



usando a estatística

$$\Delta_n(u) = \max_{a \in A} \max_{w \in B_{\mathcal{T}_n}(u)} (|\hat{p}_n(a|w) - \hat{p}_n(a|u)|).$$

Fixada uma constante  $\delta > 0$  e se  $\Delta_n(u) > \delta$  rejeitamos a hipótese nula  $H_0^{(u)}$ , isto é, mantemos  $B_{\mathcal{T}_n}(u) \in \mathcal{T}_n$ . Quando não rejeitamos  $H_0^{(u)}$ , podamos o ramo  $B_{\mathcal{T}_n}(u)$  e a nova árvore candidata será

$$\mathcal{T}_n = (\mathcal{T}_n \setminus B_{\mathcal{T}_n}(u)) \cup \{u\}.$$

A cada passo selecionamos sempre a maior sequência  $u \in A^*$  que não foi testada ainda. Repetimos esses passos até que não existam mais ramos a serem podados. Ao final obtemos uma árvore de contextos estimada  $\hat{\tau}_n$ .

É importante ressaltar que o resultado  $\hat{\tau}_n = \tau$  vale quase certamente para  $n$  suficientemente grande, ou seja, o algoritmo contexto é consistente. A demonstração desse resultado pode ser visto em (CASTRO; GALVES, 2016).

### 3.2.2 Método BIC

Uma outra maneira de estimarmos árvores probabilísticas de contextos é utilizando o método de Schwarz (SCHWARZ et al., 1978), ou critério de informação Bayesiano (BIC). Nesse método, a cada passo, comparamos duas árvores de contextos. Essas árvores são iguais a menos de um ramo, ver Figura 5. O critério de comparação entre essas duas árvores é utilizando a verossimilhança penalizada.

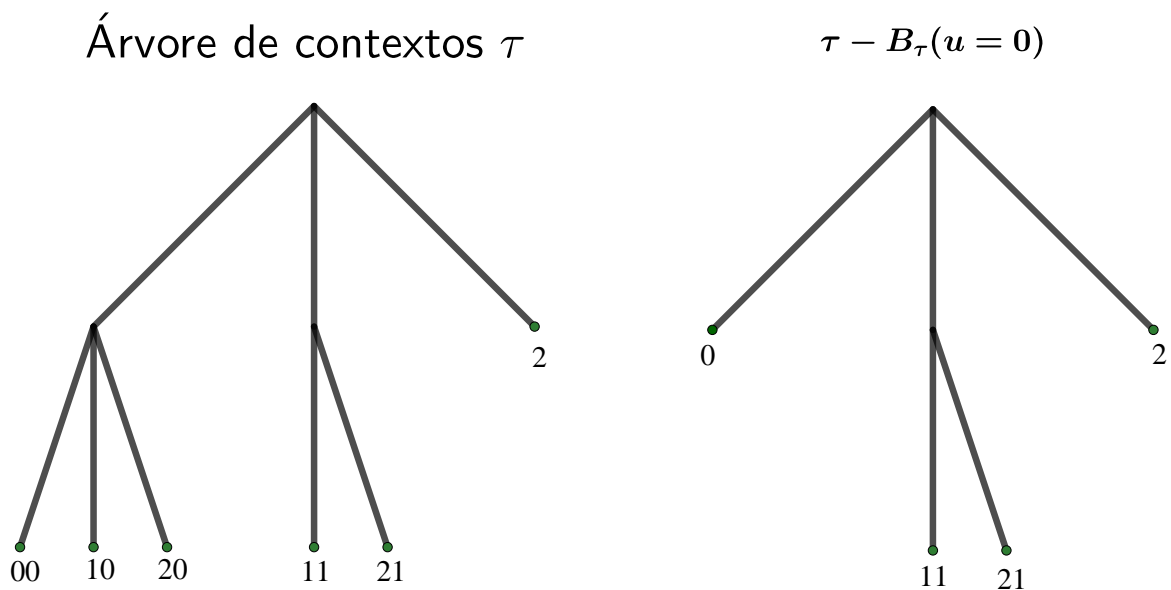


Figura 5 – Retirando ramo para aplicação do método BIC.

**Definição 3.2.3.** O estimador da árvore de contexto pelo BIC com uma constante de penalização  $c > 0$  é definido como

$$\hat{\tau}_{BIC(X_1^n; c)} = \arg \max_{\tau \in \mathcal{T}_n} \left\{ \log L_\tau(X_1^n) - c \frac{(|A| - 1)|\tau|}{2} \log n \right\}$$

em que  $|A|$  representa a cardinalidade do alfabeto da árvore  $\tau$ ,  $|\tau|$  representa a cardinalidade da árvore  $\tau$  e  $L_\tau(X_1^n) = \prod_{u \in \tau} \prod_{a \in A} p(a|u)^{N(ua)}$ .

Uma característica do método BIC é o fator de penalização. Acontece que quando passamos da profundidade correta da cadeia  $K$ , a verossimilhança continua aumentando, porém muito pouco, não vale a pena escolher uma ordem maior do que  $k$ , pois nesse caso estaríamos modelando ruído, e caindo no problema de superestimação. A explicação do porquê da penalização ser da ordem  $O(\log(n))$  é um resultado do Teorema 3.2.2 a seguir.

**Teorema 3.2.2.** Assuma  $(X_1^n)$  uma amostra de uma cadeia estocástica com memória de alcance variável compatível com a árvore probabilística de contextos  $(\tau^*, p^*)$  com  $\tau^*$  finito,  $\mathcal{L}(\tau^*) \leq K$ ,  $K \in \mathbb{N}$ . Então os resultados que seguem convergem quase certamente com  $n \rightarrow \infty$

(1) (Subestimação) Para qualquer árvore admissível  $\tau$ , com  $\tau' \prec \tau^*$ , existe uma constante  $c(\tau^*, \tau') > 0$  de tal modo que

$$\log\{L_{\tau^*}(X_1^n)\} - \log\{L_{\tau'}(X_1^n)\} \geq c(\tau^*, \tau')n$$

(2) (Superestimação) Para quaisquer árvores de contextos admissíveis  $\tau'$  e  $\tau''$ , com  $\tau^* \preceq \tau' \prec \tau''$ , existe uma constante  $c(\tau', \tau'') > 0$  de tal modo que

$$\log\{L_{\tau''}(X_1^n)\} - \log\{L_{\tau'}(X_1^n)\} \leq c(\tau', \tau'') \log(n)$$

O critério de seleção de árvores pelo método BIC é semelhante ao método do algoritmo de contextos apresentado na subseção 3.2.1. Iniciamos com a maior árvore admissível  $\mathcal{T}_n$  e para cada ramo com um número de filhos maior ou igual que 2, testamos a seguinte hipótese

$$H_0^{(u)} : p(a|ub) = p(a|u), \quad \forall a, b \in A.$$

Para cada ramo  $u$  da árvore utilizamos a seguinte estatística de teste

$$\Delta_n(u) = \sum_{b \in A} \sum_{a \in A} N(bua) \log \frac{\hat{p}_n(a|ub)}{\hat{p}_n(a|u)}.$$

Fixamos uma constante  $c > 0$  e se  $\Delta_n(u) > c' \log(n)$ , sendo  $c' = \frac{c|A|}{2}$  rejeitamos a hipótese  $H_0^{(u)}$ , mantemos  $B_{\mathcal{T}_n}(u) \in \mathcal{T}_n$ . Quando não rejeitamos  $H_0^{(u)}$ , podamos o ramo  $B_{\mathcal{T}_n}(u)$  e a nova árvore candidata será

$$\mathcal{T}_n = (\mathcal{T}_n \setminus B_{\mathcal{T}_n}(u)) \cup \{u\}.$$

Repetimos esse procedimento para cada ramo  $u$  da árvore de contexto admissível.

O estimador  $\hat{\tau}_{BIC(X_1^n; c)}$  também é consistente para  $\tau$  e uma demonstração para esse resultado pode ser visto em (CSISZÁR; TALATA, 2006).

## 4 Simulações e Resultados

Para estudarmos o comportamento das cadeias estocásticas com memória de alcance variável é muito importante realizar simulações para avaliar o desempenho dos estimadores aqui estudados, podendo também observar diferentes cenários para os parâmetros de interesse. Para tal tarefa foi implementado alguns algoritmos na linguagem R. O primeiro deles tem o objetivo de simular uma cadeia com memória de alcance variável. Para isso devemos informar os contextos da árvore, as respectivas probabilidades de transição e o tamanho da amostra desejada. Por exemplo, se desejarmos criar uma amostra da árvore  $\tau_1$  representada pela Figura 6 abaixo.

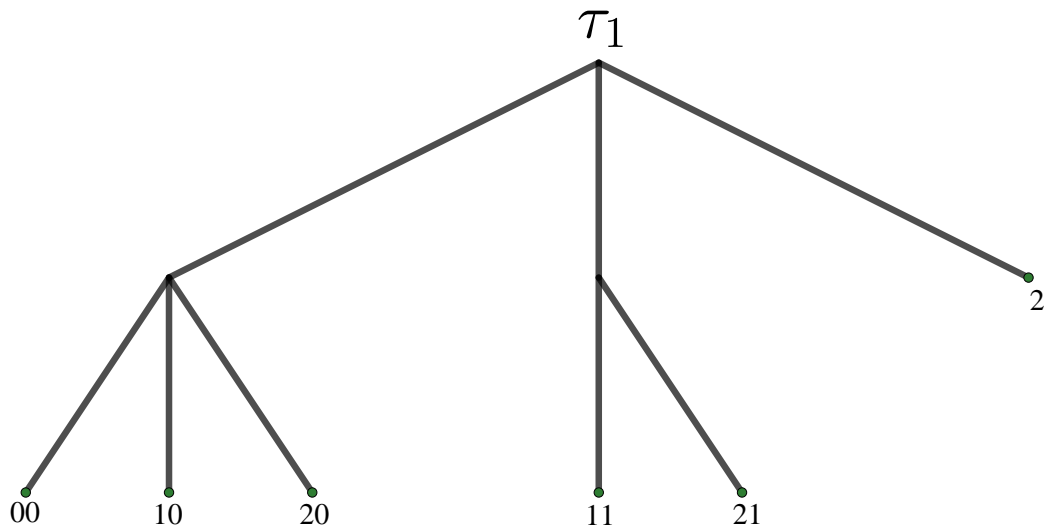


Figura 6 – Árvore de contexto  $\tau_1 = \{00, 10, 20, 11, 21, 2\}$ .

e com probabilidades de transição,

$$P = \begin{matrix} & \begin{matrix} 0 & 1 & 2 \end{matrix} \\ \begin{matrix} 00 \\ 10 \\ 20 \\ 11 \\ 21 \\ 2 \end{matrix} & \begin{pmatrix} 0.3 & 0 & 0.7 \\ 0.6 & 0 & 0.4 \\ 0.7 & 0 & 0.3 \\ 0.6 & 0.2 & 0.2 \\ 0.3 & 0.4 & 0.3 \\ 0.7 & 0.2 & 0.1 \end{pmatrix} \end{matrix}$$

devemos escrever,

```
Amostra_A(x1=c("00", "10", "20", "11", "21", "2"),
```

```

x2= c(0.3, 0, 0.7, 0.6, 0, 0.4, 0.7, 0, 0.3,0.6,
      0.2, 0.2, 0.3, 0.4, 0.3, 0.7, 0.2, 0.1),
x3=10000)

```

A construção da função *Amostra\_A* e demais funções pode ser vista no Apêndice A.

Outro algoritmo implementado foi o algoritmo contexto descrito no capítulo anterior, para seu funcionamento é necessário informar uma amostra de símbolos, a profundidade máxima da árvore de contexto a ser considerada e o nível de significância desejado. Dessa forma é possível obter uma estimativa da árvore de contextos.

## 4.1 Estimação via algoritmo contexto

Com a utilização do primeiro algoritmo implementado podemos realizar um estudo do desempenho do algoritmo contexto. Para um melhor entendimento vamos para um exemplo bem detalhado.

Seja  $X_1^{10000}$  uma amostra da árvore  $\tau_1$  que vimos anteriormente. Vamos utilizar o algoritmo de contexto para estimar os contextos da árvore  $\tau_1$ .

$x_1^{10000} = (1, 1, 1, 0, 0, 2, 0, 0, 2, 0, 0, 2, 0, 0, 0, 2, 1, 2, 0, 0, 2, 0, 2, 1, 2, 0, 2, 0, 2, 2, 0, 0, 2, 1, 2, 0, 2, 0, 0, 0, 2, 1, 2, 2, 0, 0, \dots)$

A função *Contexto* criada possui 3 entradas, a primeira trata da amostra que queremos analisar, a segunda refere-se a profundidade máxima da árvore que vamos adotar (nesse exemplo vamos adotar a profundidade máxima 3), e a última entrada devemos informar o nível de significância, para cada ramo, que adotaremos um  $\delta$ .

A função começa identificando o alfabeto da amostra fornecida,  $A = \{0, 1, 2\}$ . Com isso criamos todos os possíveis contextos com tamanho da profundidade informada, ou seja,  $\hat{\tau}_{1_0}$  é uma árvore completa de tamanho 3 e está representada pela Figura 7.

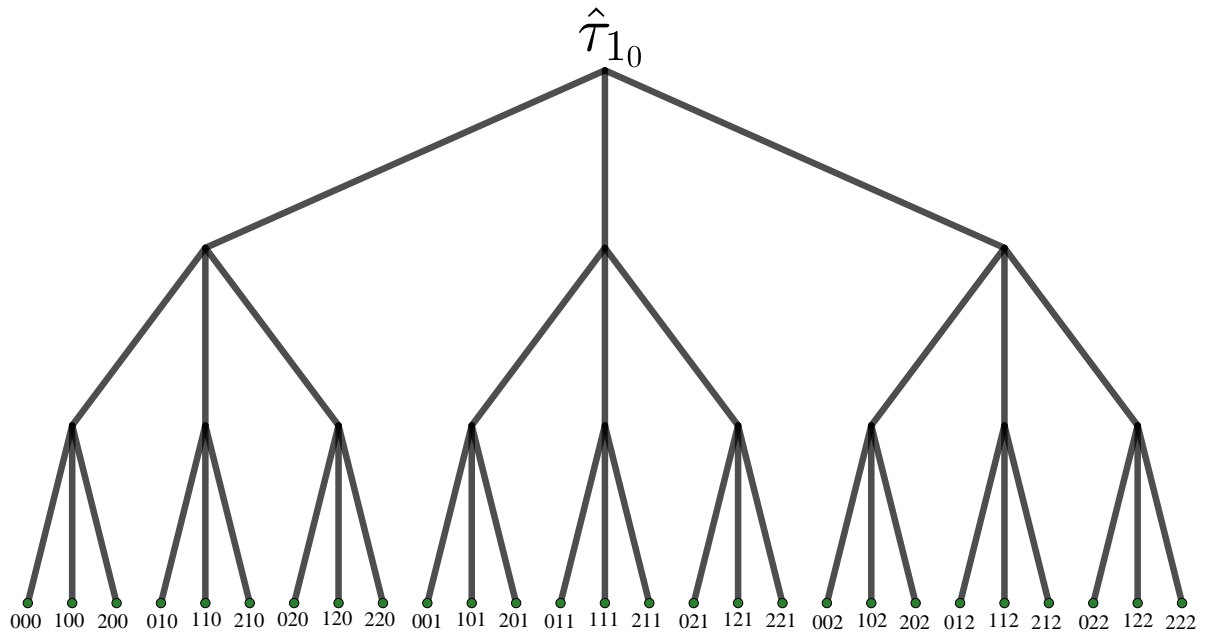


Figura 7 – Árvore de contexto  $\hat{\tau}_{10}$ .

Em seguida é calculado quantas vezes cada contexto aparece na amostra. Se o contexto não aparece ele é imediatamente desconsiderado como suposto contexto da árvore. A Tabela 1 retrata essas quantidades.

Tabela 1 – Número de vezes que um suposto contexto aparece na amostra.

	w (contexto)								
	000	100	200	010	110	210	020	120	220
N(w)	866	216	1719	0	175	199	2001	204	237
	001	101	201	011	111	211	021	121	221
N(w)	0	0	0	0	62	237	532	48	68
	002	102	202	012	112	212	022	122	222
N(w)	1935	157	723	0	63	212	282	23	39

Como tivemos contextos que não aparecem na amostra, uma nova árvore  $\hat{\tau}_{11}$  é encontrada, e com isso, temos nossa primeira árvore admissível, ver Figura 8.

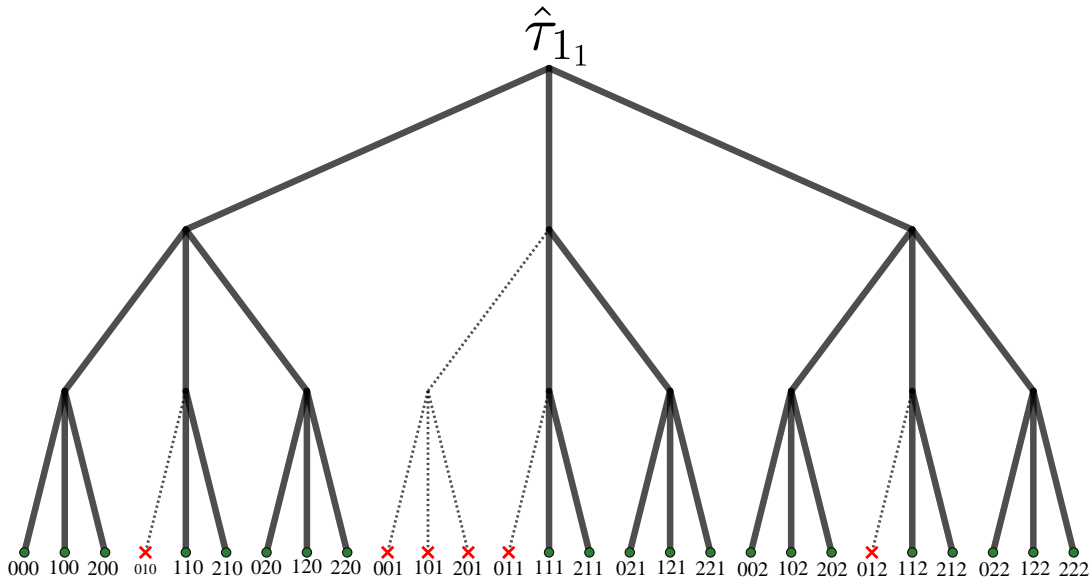


Figura 8 – Árvore de contexto admissível  $\hat{\tau}_{11}$ .

O próximo passo é testar a hipótese nula

$$H_0^{(u)} : p(a|w) = p(a|u), \quad \forall a \in A, \quad \forall w \in B_{\mathcal{T}_n}(u).$$

usando a estatística

$$\Delta_n(u) = \max_{a \in A} \max_{w \in B_{\mathcal{T}_n}(u)} (|\hat{p}_n(a|w) - \hat{p}_n(a|u)|).$$

Fixamos  $\delta = 0,1$ . Se  $\Delta_n(w) > \delta$  rejeitamos a hipótese nula  $H_0^{(u)}$ , isto é, mantemos  $B_{\mathcal{T}_n}(u) \in \mathcal{T}_n$ , caso contrário, podemos o ramo  $B_{\mathcal{T}_n}(u)$ .

Vamos começar o procedimento com o ramo  $B_{\tau_{11}n}(00)$ . Na Tabela 2 apresentamos os valores que são necessários para calcularmos a estatística do teste  $\Delta_n(00)$ .

Tabela 2 – Valores para calcular  $\Delta_n(00)$ .

$w$	$u$	$a$	$\hat{p}(a w)$	$\hat{p}(a u)$	$ \hat{p}(a w) - \hat{p}(a u) $
000	00	0	0,326	0,309	<b>0,017</b>
000	00	1	0,000	0,000	0,000
000	00	2	0,674	0,691	<b>0,017</b>
100	00	0	0,319	0,309	0,010
100	00	1	0,000	0,000	0,000
100	00	2	0,681	0,691	0,010
200	00	0	0,300	0,309	0,009
200	00	1	0,000	0,000	0,000
200	00	2	0,700	0,691	0,009

Obtemos  $\Delta_n(00) = 0,017$ , como  $\Delta_n(00) < \delta$  não rejeitamos a hipótese nula, ou seja, devemos podar o ramo  $B_{\tau_{11}n}(00)$ .

Realizando o mesmo procedimento para todas as sequências  $u$ , com  $\ell(u) = 2$ , obtemos a Tabela 3 abaixo.

Tabela 3 – Resultado dos testes de hipótese para as sequências de  $u$ , com  $\ell(u) = 2$ .

$u$	$\Delta_n(u)$	$H_0^{(u)}$	$B_{\tau_{1,n}}(u)$
00	0,017	não rejeita	podamos
10	0,014	não rejeita	podamos
20	0,025	não rejeita	podamos
11	0,126	rejeita	não podamos
21	0,018	não rejeita	podamos
02	0,014	não rejeita	podamos
12	0,095	não rejeita	podamos
22	0,092	não rejeita	podamos

De todas as sequências testadas apenas a  $u=11$  rejeita a hipótese nula. Com isso apenas o ramo  $B_{\tau_{1,n}}(11)$  não é podado. A Figura 9 mostra com fica a nova árvore admissível  $\hat{\tau}_{1_2}$

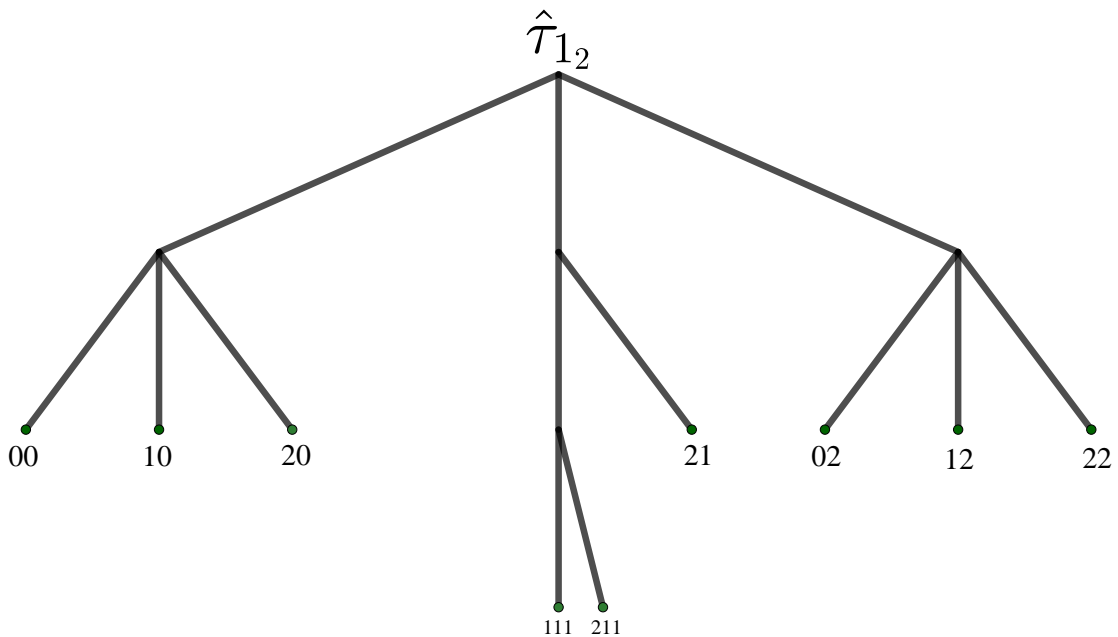


Figura 9 – Nova Árvore de contexto admissível  $\hat{\tau}_{1_2}$ .

A ideia agora é aplicar o mesmo procedimento para os ramos induzidos por sequências  $u$  de comprimento 1. Observe que  $\{u = 1\}$  não deverá ser testada pois o ramo  $B_{\tau_{1,n}}(11)$  não foi podado. Na Tabela 4 apresentamos os resultados dos testes para essas sequências.



Tabela 4 – Resultado dos testes de hipótese para as sequências  $u$ , com  $\ell(u) = 1$ .

$u$	$\Delta_n(u)$	$H_0^{(u)}$	$B_{\tau_{1,2^n}}(u)$
0	0,205	rejeita	não podamos
2	0,031	não rejeita	podamos

Com esse resultado chegamos na árvore estimada  $\hat{\tau}_n$  que não terá o ramo  $B_{\tau_{1,2^n}}(2)$ . A Figura 10 compara a árvore original  $\tau_1$ , que foi gerado a amostra, com a árvore estimada  $\hat{\tau}_{1_n}$  pelo algoritmo contexto.

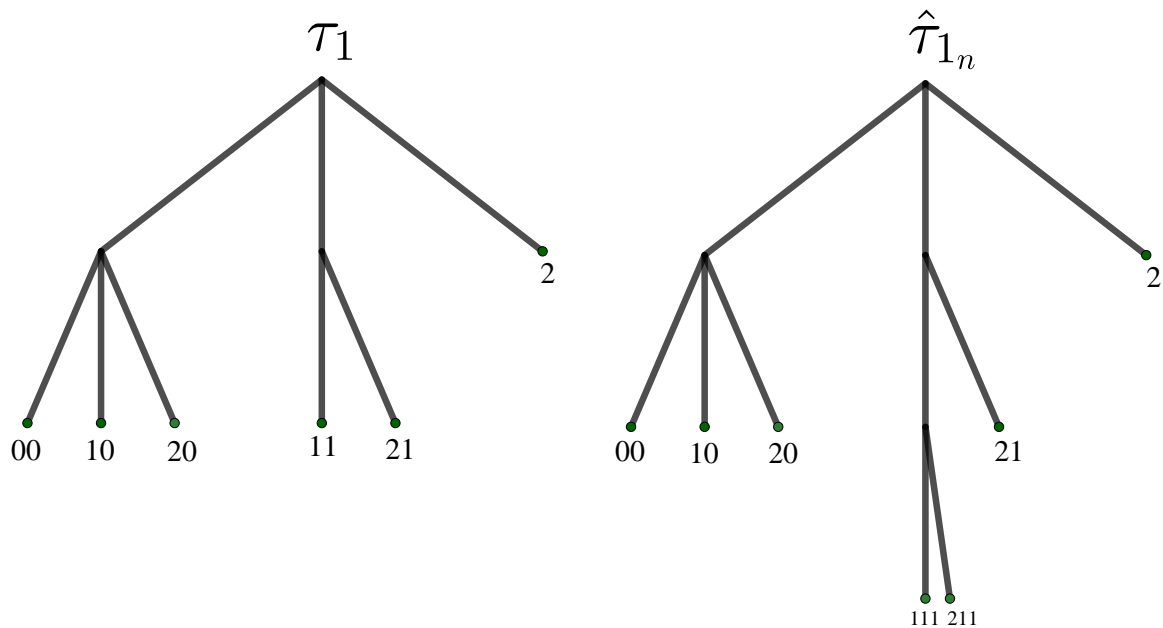


Figura 10 – Comparação da árvore  $\tau_1$  geradora da amostra com a árvore estimada  $\hat{\tau}_{1_n}$  pelo algoritmo de contexto.

A estimativa pelo algoritmo de contexto errou por apenas um ramo. Aplicar esse algoritmo para apenas uma amostra não nos fornece o seu desempenho, a ideia agora é utilizar réplicas de Monte Carlos para termos uma noção de como é a tendência e a variabilidade desse estimador, o método de Monte Carlo pode ser visto em (SHREIDER; BUSLENKO, 1966). Para isso vamos gerar 100 réplicas da mesma árvore com tamanho 10000, e em seguida estimar cada árvore pelo método do algoritmo contexto. Detalhes dessa e das outras simulações pode ser encontrado no Apêndice B. O resultado dessas estimativas pode ser visto na Figura 11.

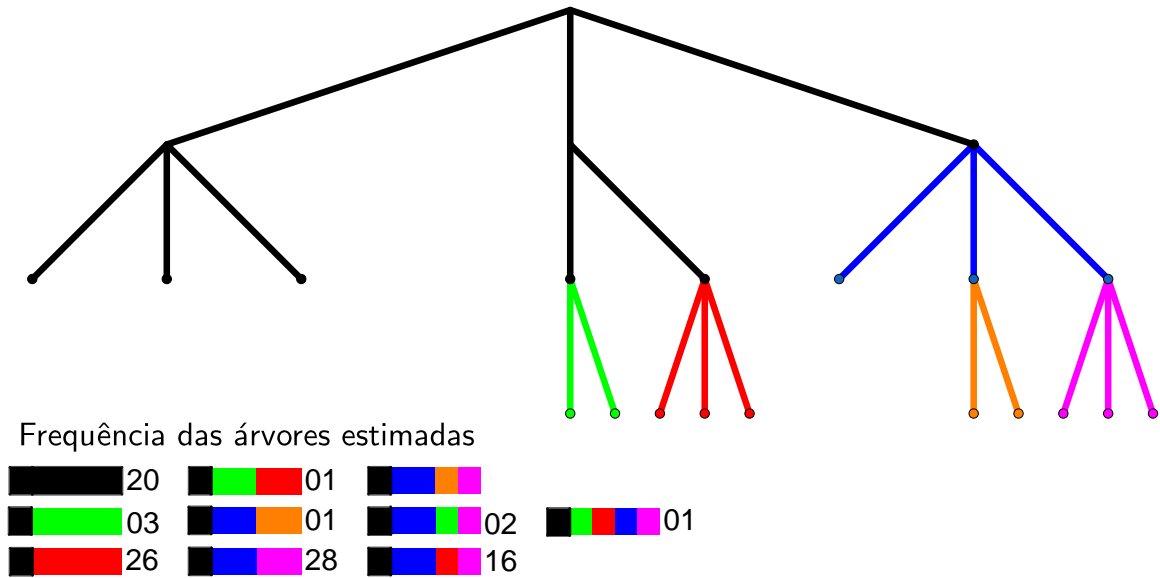


Figura 11 – Distribuição das 100 árvores estimadas.

A Figura 11 mostra que das 100 amostras geradas houve um total de 10 árvores distintas estimadas, sendo que 4 delas correspondem a 90% das amostras. Observamos que o algoritmo contexto estimou corretamente a árvore 20 vezes, ficando atrás da árvore que inclui o ramo  $\{B_{\tau_n}(2) \cup B_{\tau_n}(22)\}$  (estimada 28 vezes) e da árvore que inclui o ramo  $B_{\tau_n}(21)$  (estimada 26 vezes).

Podemos questionar se o desempenho do estimador  $\hat{\tau}_n$  pelo algoritmo contexto foi satisfatório, visto que se trata de um estimador consistente e geramos amostras de tamanho 10000. A Tabela 1 apresenta um indício para o desempenho não ter sido tão bom quanto se esperava. Sabemos que  $B_{\tau_n}(22) = \{022, 122, 222\}$  e pela Tabela 1 podemos notar que esse conjunto apresenta as duas sequências com menor valor da função  $N_n(\cdot)$ ,  $N_n(122) = 23$  e  $N_n(222) = 39$ . Esses valores baixos podem comprometer o desempenho do estimador  $\hat{p}_n(a|u)$  que é utilizado para gerar  $\hat{\tau}_n$ . O mesmo problema ocorre se observarmos o ramo  $B_{\tau_n}(21) = \{021, 121, 221\}$  que apresenta a terceira sequência com menor valor de  $N_n(\cdot)$ ,  $N_n(121) = 48$ .

Para confirmar que se essa suspeita, de fato, faz sentido, vamos gerar novamente 100 amostras, agora com o tamanho 20000. O procedimento adotado é exatamente o mesmo; a Figura 12 apresenta as árvores estimadas com essa nova quantidade de símbolos.

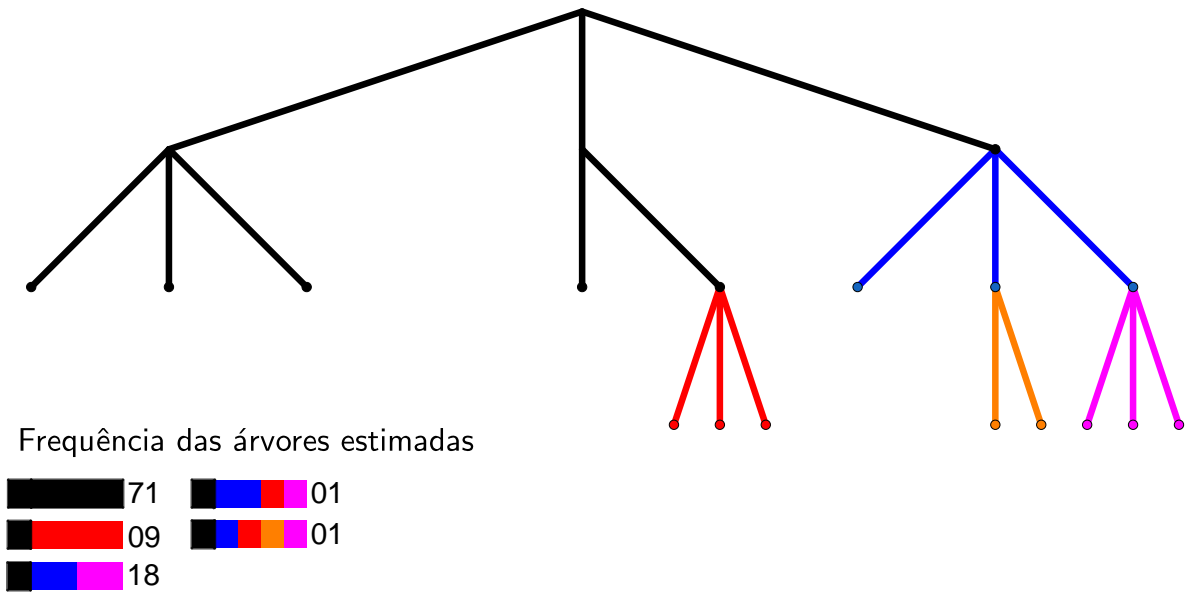


Figura 12 – Distribuição das 100 árvores estimadas com o dobro de símbolos.

A Figura 12 apresenta um resultado bem melhor que a Figura 11, temos que o número de árvores estimadas foi reduzido pela metade, indicando uma menor variabilidade. E desta vez a árvore que foi estimada mais vezes é justamente a árvore que gerou a amostra. Ou seja, a suspeita que levantamos anteriormente parece de fato ter influenciado o desempenho do estimador pelo algoritmo contexto.

## 4.2 Estimação via método BIC

A utilização do método BIC é bem semelhante ao algoritmo contexto, basicamente o que se altera é a estatística do teste de hipótese. Considerando a mesma amostra do exemplo anterior, a primeira árvore admissível é dada por  $\hat{\tau}_{1_1}$  que é mostrada pela Figura 8. Agora devemos calcular a estatística

$$\Delta_n(u) = \sum_{b \in A} \sum_{a \in A} N(bua) \log \frac{\hat{p}_n(a|ub)}{\hat{p}_n(a|u)}.$$

Fixando uma constante  $c' = 2$ , se  $\Delta_n(u) > 2 \cdot \log(10000) \implies \Delta_n(u) > 18,421$  rejeitamos a hipótese nula  $H_0^{(u)}$ , ou seja, mantemos  $B_{\mathcal{T}_n}(u) \in \mathcal{T}_n$ . Caso contrário podamos o ramo  $B_{\mathcal{T}_n}(u)$ .

Vamos realizar o cálculo dessa estatística para o ramo  $B_{\tau_{1_1 n}}(00)$ . Na Tabela 5 a seguir apresentamos os valores que são necessários para este cálculo.

Tabela 5 – Valores para calcular  $\Delta_n(00)$  pelo método BIC.

$u$	$a$	$b$	$N(bua)$	$\hat{p}(a ub)$	$\hat{p}(a u)$	$N(bua) \cdot \log[\hat{p}(a ub)/\hat{p}(a u)]$
00	0	0	282	0,326	0,309	15,103
00	1	0	0	0,000	0,000	-
00	2	0	584	0,674	0,691	-14,547
00	0	1	69	0,319	0,309	2,198
00	1	1	0	0,000	0,000	-
00	2	1	147	0,681	0,691	-2,143
00	0	2	515	0,300	0,309	-15,223
00	1	2	0	0,000	0,000	-
00	2	2	1204	0,700	0,691	15,580

Obtemos  $\Delta_n(00) = 0,968$ , e como  $\Delta_n(00) < 18,421$  não rejeitamos a hipótese nula, ou seja, devemos podar o ramo  $B_{\tau_{1n}}(00)$ .

Realizando o mesmo procedimento para todas as sequências  $u$ , com  $\ell(u) = 2$ , obtemos a Tabela 6 seguinte.

Tabela 6 – Resultado dos testes de hipótese para as sequências  $u$ , com  $\ell(u) = 2$  pelo método BIC.

$u$	$\Delta_n(u)$	$H_0^{(u)}$	$B_{\tau_{1n}}(u)$
00	0,968	não rejeita	podamos
10	0,112	não rejeita	podamos
20	0,421	não rejeita	podamos
11	4,360	não rejeita	podamos
21	0,099	não rejeita	podamos
02	0,712	não rejeita	podamos
12	2,834	não rejeita	podamos
22	1,910	não rejeita	podamos

Diferentemente do algoritmo contexto, o método BIC poda o ramo  $B_{\tau_{1n}}(11)$ , fazendo com que tenhamos uma nova árvore admissível, representada pela Figura 13.

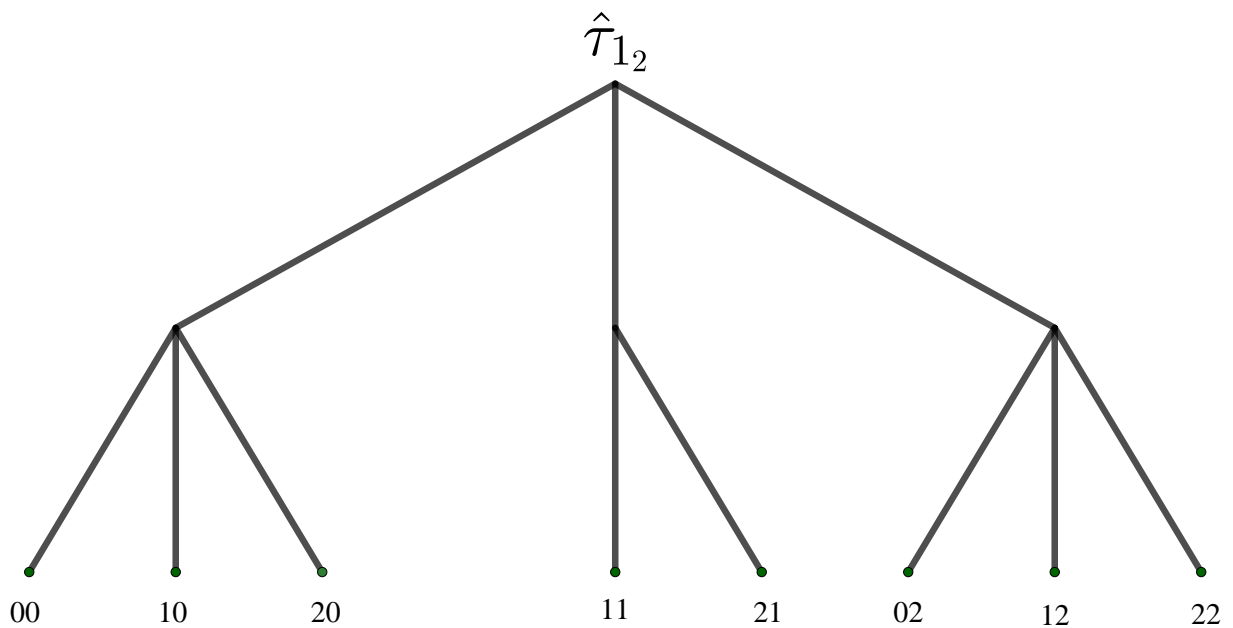


Figura 13 – Nova Árvore de contexto admissível  $\hat{\tau}_{12}$  pelo método BIC.

Aplicando o mesmo procedimento para os ramos induzidos por sequências  $u$  de comprimento 1, obtemos a Tabela 7.

Tabela 7 – Resultado dos testes de hipótese para as sequências  $u$ , com  $\ell(u) = 1$  pelo método BIC.

$u$	$\Delta_n(u)$	$H_0^{(u)}$	$B_{\tau_{12,n}}(u)$
0	423,086	rejeita	não podemos
1	32,628	rejeita	não podemos
2	1,207	não rejeita	podamos

Com esse resultado chegamos na árvore estimada  $\hat{\tau}_n$  que será igual a árvore que gerou a amostra. Para termos uma noção do desempenho desse estimador pelo método BIC utilizaremos as mesmas 100 amostras que geramos para o algoritmo contexto, no caso em que cada amostra possui 10000 símbolos o resultado obtido é de uma precisão de estimação de 100%, ou seja, o método BIC estimou corretamente em todas as 100 amostras. No método anterior, como o resultado não tinha sido tão bom, resolvemos dobrar a quantidade de símbolos. Como neste método o resultado foi excelente, vamos reduzir o tamanho da amostra para 5000.

O Resultado obtido também foi muito bom, ao todo apenas duas árvores foram estimadas, sendo a correta estimada 82 vezes e uma outra árvore 18 vezes; a Figura 14 mostra as duas árvores estimadas.

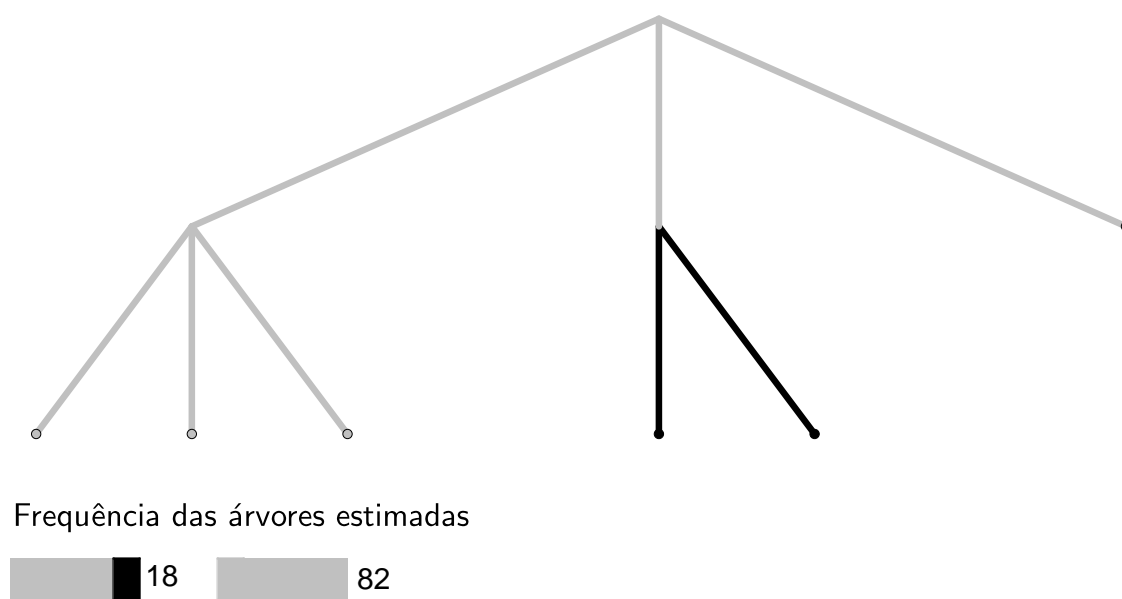


Figura 14 – Distribuição das 100 árvores estimadas com a metade de símbolos pelo método BIC.

Neste caso, tivemos a subestimação da árvore, o método BIC sugeriu a retirada do o ramo  $B_{\mathcal{T}_n}(1)$  quando na verdade isso não deveria ter acontecido. Um fato interessante de se observar é que pelo método do algoritmo contexto só ocorreu superestimação da árvore.

## 5 Considerações Finais

Neste trabalho, foi apresentado conceitos, propriedades e o estimador de máxima verossimilhança sobre as cadeias de Markov, que é fundamental para introduzir a ideia das cadeias estocásticas com memória de alcance variável. Em seguida, alguns conceitos pertinentes para o entendimento dessas cadeias foram expostos e exemplificados com situações e ilustrações.

O maior interesse deste trabalho foi estudar dois critérios de seleção para árvores de contextos, o algoritmo contexto e o método BIC. Para este estudo foi fundamental a implementação de funções e algoritmos no software R, pois, com isso, podemos simular amostras desses tipos de cadeias e comparar o desempenho desses dois critérios na hora da estimação.

Concluimos que, o método BIC apresentou melhores resultados quando comparado com o algoritmo contexto. Para essa comparação foi fixado uma estrutura de árvore com alfabeto de tamanho 3, foi escolhido aleatoriamente probabilidades de transição e a princípio amostras de tamanho 10000. O que podemos notar foi que esse tamanho não foi suficiente para que o algoritmo contexto apresentasse bom desempenho, que pode ser explicado pela estrutura da árvore escolhida, que torna a aparição de algumas sequências um tanto raras. Já o método BIC não sofreu com esse problema, apresentando um bom desempenho.

Apesar do método BIC ter se destacado, vale ressaltar alguns pontos importantes. Para aplicação do mesmo, tivemos que escolher uma constante arbitrária  $c = 2$ , entretanto o ideal seria um estudo dos melhores valores para essa constante, por exemplo como apresentado em (GALVES et al., 2012).

Por fim, as cadeias estocásticas com memória de alcance variável, é um bom método de generalização quando precisamos observar um passado de tamanho maior do que 1 sem precisar utilizar uma cadeia de Markov de alcance  $k$ . Neste modelo podemos variar outros parâmetros como a estrutura da árvore, probabilidades de transição, profundidade, tamanho do alfabeto e outros.

# Referências

- BÜHLMANN, P.; WYNER, A. J. et al. Variable length markov chains. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 27, n. 2, p. 480–513, 1999. 11
- BUSCH, J. R. et al. Testing statistical hypothesis on random trees and applications to the protein classification problem. *The Annals of Applied Statistics*, Institute of Mathematical Statistics, v. 3, n. 2, p. 542–563, 2009. 12
- CASTRO, B. M. d.; GALVES, J. A. Processos estocásticos conduzidos por cadeias com memória de alcance variável e o jogo do goleiro. 2016. 24
- CSISZÁR, I.; TALATA, Z. Context tree estimation for not necessarily finite memory processes, via bic and mdl. *IEEE Transactions on Information theory*, IEEE, v. 52, n. 3, p. 1007–1016, 2006. 26
- GALVES, A. et al. Context tree selection and linguistic rhythm retrieval from written texts. *The Annals of Applied Statistics*, Institute of Mathematical Statistics, v. 6, n. 1, p. 186–209, 2012. 11, 12, 38
- GALVES, A.; LÖCHERBACH, E. Stochastic chains with memory of variable length. *arXiv preprint arXiv:0804.2050*, 2008. 11
- HAMILTON, J. D. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, JSTOR, p. 357–384, 1989. 13
- PRATAS, D. et al. An alignment-free method to find and visualise rearrangements between pairs of dna sequences. *Scientific reports*, Nature Publishing Group, v. 5, p. 10203, 2015. 13
- R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2019. Disponível em: <<https://www.R-project.org/>>. 12
- RISSANEN, J. A universal data compression system. *IEEE Transactions on information theory*, IEEE, v. 29, n. 5, p. 656–664, 1983. 11, 23
- SCHWARZ, G. et al. Estimating the dimension of a model. *The annals of statistics*, Institute of Mathematical Statistics, v. 6, n. 2, p. 461–464, 1978. 24
- SHREIDER, I. A.; BUSLENKO, N. P. *The Monte Carlo Method of Statistical Trials*. [S.l.]: Pergamon Press, 1966. 32



## APÊNDICE A – Funções criadas no R

A função *Alf\_C* retorna o alfabeto de um contexto e possui apenas um argumento, que é o contexto, de modo que ele esteja em um único elemento do vetor.

```
Alf_C <- function(x){
  y <- sapply(1:nchar(x),function(i) substr(x,i,i))
  y <- unique(y)
  y <- sort(y)
  return(y)
}
```

Exemplo:

```
> Alf_C(c("002"))
[1] "0" "2"
```

A função *Alf\_A* retorna o alfabeto de uma árvore de contextos e possui apenas um argumento, que é a árvore, de modo que ela esteja com um conjunto de contextos, ou seja, cada elemento do vetor é um contexto.

```
Alf_A <- function(x){
  y <- list()
  for (j in 1:length(x)){
    y[[j]] <- Alf_C(x[j])
  }
  y <- unlist(y)
  y <- unique(y)
  y <- sort(y)
  return(y)
}
```

Exemplo:

```
> Alf_A(c("00", "10", "20", "1", "2"))
[1] "0" "1" "2"
```

A função *Quebra* retorna os símbolos do contexto de forma que cada um fique em uma posição do vetor, possui apenas um argumento, que é o contexto.

```
Quebra <- function(x){
  y <- sapply(1:nchar(x),function(i) substr(x,i,i))
```

```

  return(y)
}

```

Exemplo:

```

> Quebra(c("122"))
[1] "1" "2" "2"

```

A função *Amostra\_A* retorna uma amostra de símbolos, ela possui 3 argumentos,  $x_1$  que representa a árvore,  $x_2$  que representa a matriz de probabilidades de transição de um contexto para um símbolo, que deverá ser apresentada por um vetor e  $x_3$  o tamanho da amostra (valor padrão 10000).

```

Amostra_A <- function(x1, x2, x3=10000){

  TamanhoA <- x3
  P_trans <- x2
  Arvore <- x1

  MatrizT <- matrix(P_trans, nrow = length(Arvore),
                    byrow = T, dimnames = list(Arvore, Alf_A
(Arvore)))

  Amostra <- rep(NA, TamanhoA)

  ContextoI <- Quebra(sample(Arvore,1))

  Amostra[1:length(ContextoI)] <- ContextoI

  for (j in (length(ContextoI)+1):TamanhoA ) {
    i = 1
    while(all(Quebra(Arvore[i]) ==
                rev(rev(na.omit(Amostra))[1:length(Quebra(
Arvore[i]))])) == F){
      i <- i+1
    }
    a <- runif(1)
    if (a < MatrizT[i,1]){
      Amostra[j] <- Alf_A(Arvore)[1]
    }else{
      k <- 2
      while( ((a > sum(MatrizT[i,(1:(k-1))])) &

```

```

        (a < sum(MatrizT[i,(1:k)]))) == F){
      k <- k+1
    }
    Amostra[j] <- Alf_A(Arvore)[k]
  }
}

return(Amostra)
}

```

Exemplo:

```

Amostra_A(x1=c("00", "10", "20", "11", "21", "2"),
          x2= c(0.3, 0, 0.7, 0.6, 0, 0.4, 0.7, 0, 0.3,0.6,
                0.2, 0.2, 0.3, 0.4, 0.3, 0.7, 0.2, 0.1),
          x3=10)
[1] "0" "0" "0" "2" "0" "0" "2" "0" "0" "0"

```

A função *N* retorna quantas vezes um Contexto aparece em uma amostra, possui 2 argumentos, *x1* que representa a amostra e *x2* que representa o contexto de interesse de modo que cada símbolo esteja em uma posição do vetor.

```

N <- function(x1, x2){

  n <- length(x1)
  nc <- length(x2)
  aux <- 0

  for (i in 1:(n-(nc-1))) {
    if(all(x1[i:(length(x2)+i-1)] == x2)){
      aux <- aux + 1
    }
  }
  return(aux)
}

```

Exemplo:

```

> N(x1=c("0", "0", "0", "2", "0", "0", "2", "0", "0", "0"), x2=c
    ("0", "0"))
[1] 5

```

A função  $N_p$  retorna um vetor com as estimativas das probabilidades de transição de um contexto para os símbolos da amostra, ela possui 2 argumentos,  $x1$  representa a amostra e  $x2$  que representa o contexto de interesse de modo que cada símbolo esteja em uma posição do vetor.

```
N_p <- function(x1, x2){
  n <- length(x1)
  a <- Alf_A(x1)
  aux <- rep(0,length(a))
  for (j in 1:length(a)) {
    aux[j] <- N(x1, c(x2,a[j]))/N(x1, x2)
  }
  return(aux)
}
```

Exemplo:

```
> N_p(x1=c("0","0","0","2","0","0","2","0","0","2"), x2=c
      ("0","0"))
[1] 0.25 0.75
```

A função *Contexto* retorna os contextos de uma amostra de uma árvore de contextos pelo método do algoritmo contexto, possui 3 argumentos,  $x1$  representa a amostra,  $x2$  representa a profundidade máxima da árvore e  $x3$  o valor de  $\delta$  (valor padrão 0,1).

```
Contexto <- function(x1, x2, x3=0.1){
  require(gtools)
  a <- Alf_A(x1)

  lista <- list()
  for (n in 1:x2) {
    lista[[n]] <- permutations(length(a), n, repeats.allowed
    = T, a)[,c(n:1),drop = FALSE]
  }

  for (i in length(a):1) {
    for (j in 1:(length(a)^i)) {
      if(N(x1, lista[[i]][j,]) == 0){
        lista[[i]][j,] <- NA
      }
    }
  }
}
```

```

}

nalf <- length(a)
for (i in x2:2) {
  for (j in 1:nalf^(i-1)) {
    lr <- lista[[i]][(nalf*(j-1)+1):(nalf*j),]
    if (is.na(lista[[i-1]][j]) == F){
      if(sum(is.na(lista[[i]][(nalf*(j-1)+1):(nalf*j),]))
== (nalf*i - i)){
        lista[[i]][(nalf*(j-1)+1):(nalf*j),] <- NA
      }
      aux <- rep(0,nalf)
      for (k in 1:nalf) {
        if(all(is.na(lr[k,])) == F){
          aux[k] <- max(abs(N_p(x1,lr[k,]) - N_p(x1,lista
[[i-1]][j,])), na.rm=T)
        }
      }
      if(max(aux, na.rm = T) < x3){
        lista[[i]][(nalf*(j-1)+1):(nalf*j),] <- NA
      }else{
        l <- i
        m <- j
        while (l > 1) {
          l<-l-1
          lista[[l]][m,] <- NA
          r <- -m%%nalf
          m <- ((m+r)/nalf)
        }
      }
    }
  }
}

contexto <- c()
aux <- 1

```

```

for (i in 1:x2) {
  for (j in 1:(nalf^i)) {
    if (all(is.na(lista[[i]][j,])) == F){
      contexto[aux] <- paste0(lista[[i]][j,], collapse =
"")
      aux <- aux + 1
    }
  }
}

return(contexto)
}

```

Exemplo:

```

> Amostra <- Amostra_A(x1=c("00", "10", "20", "11", "21",
"2"),
                      x2= c(0.3, 0, 0.7, 0.6, 0, 0.4, 0.7, 0,
0.3,0.6,
                      0.2, 0.2, 0.3, 0.4, 0.3, 0.7,
0.2, 0.1),
                      x3=10000)
> Contexto(x1=Amostra, x2=3, x3=0.1)
[1] "2" "00" "10" "20" "21" "111" "211"

```

A função *Contexto\_BIC* retorna os contextos de uma amostra de uma árvore de contextos pelo método BIC, possui 3 argumentos,  $x1$  representa a amostra,  $x2$  representa a profundidade máxima da árvore e  $x3$  o valor de  $c$  (valor padrão 2).

```

Contexto_BIC <- function(x1, x2, x3=2){
  require(gtools)
  a <- Alf_A(x1)

  lista <- list()
  for (n in 1:x2) {
    lista[[n]] <- permutations(length(a), n, repeats.allowed
= T, a)[,c(n:1),drop = FALSE]
  }

  for (i in length(a):1) {
    for (j in 1:(length(a)^i)) {
      if(N(x1, lista[[i]][j,]) == 0){

```

```

        lista[[i]][j,] <- NA
    }

}

}

nalf <- length(a)
for (i in x2:2) {
    for (j in 1:nalf^(i-1)) {
        lr <- lista[[i]][(nalf*(j-1)+1):(nalf*j),]
        if (is.na(lista[[i-1]][j]) == F){
            if(sum(is.na(lista[[i]][(nalf*(j-1)+1):(nalf*j),]))
== (nalf*i - i)){
                lista[[i]][(nalf*(j-1)+1):(nalf*j),] <- NA
            }
            aux <- rep(0,nalf)
            aux2 <- NULL
            for (k in 1:nalf) {
                if(all(is.na(lr[k,])) == F){
                    for (l in 1:nalf) {
                        aux2[l] <- N(x1, c(lr[k,],a[l]))
                    }
                    aux[k] <- sum(aux2 * log(N_p(x1,lr[k,])/N_p(x1,
lista[[i-1]][j,])), na.rm = T)
                }
            }
            if(sum(aux) < x3*log(length(x1))){
                lista[[i]][(nalf*(j-1)+1):(nalf*j),] <- NA
            }else{
                l <- i
                m <- j
                while (l > 1) {
                    l<-l-1
                    lista[[l]][m,] <- NA
                    r <- -m%%nalf
                    m <- ((m+r)/nalf)
                }
            }
        }
    }
}

```

```

    }
  }
}

contexto <- c()
aux <- 1
for (i in 1:x2) {
  for (j in 1:(nalf^i)) {
    if (all(is.na(lista[[i]][j,])) == F){
      contexto[aux] <- paste0(lista[[i]][j,], collapse =
"")
      aux <- aux + 1
    }
  }
}

return(contexto)
}

```

Exemplo:

```

> Amostra <- Amostra_A(x1=c("00", "10", "20", "11", "21",
"2"),
                      x2= c(0.3, 0, 0.7, 0.6, 0, 0.4, 0.7, 0,
0.3,0.6,
                      0.2, 0.2, 0.3, 0.4, 0.3, 0.7,
0.2, 0.1),
                      x3=10000)
> Contexto_BIC(x1=Amostra, x2=3, x3=2)
[1] "2" "00" "10" "20" "11" "21"

```



## APÊNDICE B – Simulações feitas no R

Simulação dos dados que resultaram na Figura 11.

```
#####
set.seed(9021997)
AMOSTRAS <- list()

for (i in 1:100) {
  AMOSTRAS[[i]] <- Amostra_A(x1=c("00", "10", "20", "11",
    "21", "2"),
                             x2= c(0.3, 0, 0.7, 0.6, 0, 0.4,
    0.7, 0, 0.3,0.6,
                             0.2, 0.2, 0.3, 0.4, 0.3,
    0.7, 0.2, 0.1),
                             x3=10000)
}

CONTEXTOS <-list()

for (i in 1:100) {
  CONTEXTOS[[i]] <- Contexto(AMOSTRAS[[i]], x2=3)
}

tbcontextos <- cbind(
  data.frame(id=I(unique(CONTEXTOS))),
  count=as.vector(table(match(CONTEXTOS,unique(CONTEXTOS))))
)

tbcontextos

tbcontextos$id
#####
```

Simulação dos dados que resultaram na Figura 12.

```
#####
set.seed(9021997)
AMOSTRAS2 <- list()
```

```

for (i in 1:100) {
  AMOSTRAS2[[i]] <- Amostra_A(x1=c("00", "10", "20", "11",
    "21", "2"),
    x2= c(0.3, 0, 0.7, 0.6, 0, 0.4,
    0.7, 0, 0.3,0.6,
    0.2, 0.2, 0.3, 0.4, 0.3,
    0.7, 0.2, 0.1),
    x3=20000)
}

CONTEXTOS2 <-list()

for (i in 1:100) {
  CONTEXTOS2[[i]] <- Contexto(AMOSTRAS2[[i]], x2=3)
}

tbcontextos2 <- cbind(
  data.frame(id=I(unique(CONTEXTOS2))),
  count=as.vector(table(match(CONTEXTOS2,unique(CONTEXTOS2))
  ))
)

tbcontextos2

tbcontextos2$id
#####

Simulação dos dados para o método BIC com amostras de tamanho 10000

#####
CONTEXTOS_BIC <-list()
for (i in 1:100) {
  CONTEXTOS_BIC[[i]] <- Contexto_BIC(AMOSTRAS[[i]], x2=3)
}

tbcontextos_bic <- cbind(
  data.frame(id=I(unique(CONTEXTOS_BIC))),
  count=as.vector(table(match(CONTEXTOS_BIC,unique(CONTEXTOS
  _BIC))))
)

```

)

```
tbcontextos_bic
```

```
#####
```

Simulação dos dados para o método BIC com amostras de tamanho 5000 que resultou na Figura 14.

```
#####
```

```
CONTEXTOS_BIC2 <-list()
```

```
for (i in 1:100) {
```

```
  CONTEXTOS_BIC2[[i]] <- Contexto_BIC(AMOSTRAS2[[i  
  ]][1:5000], x2=3)
```

```
}
```

```
tbcontextos3 <- cbind(  
  data.frame(id=I(unique(CONTEXTOS_BIC2))),  
  count=as.vector(table(match(CONTEXTOS_BIC2,unique(  
    CONTEXTOS_BIC2))))
```

```
)
```

```
tbcontextos3
```

```
#####
```