Departamento de Estatística - Universidade de Brasília

RAFAEL RIBEIRO BORGES

Estudo da Umidade Relativa do Ar em Brasília: Uma Aplicação de Cadeias de Ordem Variável

Brasília
2018

RAFAEL RIBEIRO BORGES

Estudo da Umidade Relativa do Ar em Brasília: Uma Aplicação de Cadeias de Ordem Variável

Projeto apresentado para obtenção do título de Bacharel em Estatística.

Departamento de Estatística - Universidade de Brasília

Orientador: Prof. Dr. Lucas Moreira

Brasília
2018

Sumário

Lista de tabelas

Lista de ilustrações

Introdução

A motivação original deste trabalho foi estudar as Cadeias de Ordem Variável e propor um modelo para tentar prever o nível de umidade relativa do ar no próximo dia. Para tal, levamos em consideração o Algoritmo BIC de Csiszar e Talata (2005) e o Algoritmo Contexto Modificado de Galves e Leonardi (2008).

Na teoria de probabilidade um processo de Markov é um processo estocástico que satisfaz a propriedade de Markov. A ideia é que é possível fazer previsões para o futuro do processo apenas com base no estado presente, independentemente dos estados passados. Pode ser chamado então de Cadeia de Markov de ordem um. Para um processo de Markov de ordem k, considera-se a sequência de k estados anteriores para a previsão do próximo estado.

O conceito de Cadeias de Ordem Variável foi introduzido por Rissanen (1983) como uma generalização dos modelos de Markov. Rissanen propôs um modelo considerando as dependências estruturais do processo onde o comprimento da porção relevante do passado (chamado de contexto) é função do próprio passado, ou seja, o número de estados atrás que se olha para determinar a probabilidade de um evento futuro não é fixo. Como nenhum contexto pode ser sufixo de outro contexto, podemos então representar o conjunto de contextos como uma árvore de probabilidades. Esses modelos são conhecidos como Cadeias de Ordem Variável e são comumente utilizados para descrever processos estocásticos uma vez que são mais flexíveis e econômicos no número de parâmetros do que as cadeias de Markov.

Além de introduzir o conceito de Cadeias de Ordem Variável, Rissanen (1983) propôs um algoritmo para estimar as árvores de contexto chamado de Algoritmo Contexto.

Diversos outros estudos abordaram a questão da estimação da árvore de contextos nas Cadeias de Ordem Variável bem como as probabilidades de transição estimadas utilizando variações baseadas no Algoritmo Contexto do Rissanen (1983). Dentre eles se destacam Buhlmann e Wyner (2003) para a ordem não limitada, o BIC de Csiszar e Talata (2005) e também Duarte et al. (2006) que deram um aprimoramento para a velocidade de convergência do Algoritmo Contexto para as cadeias de Ordem variável não limitadas.

Nos Modelos de Contaminação Estocástica, é assumido que o processo original está contaminado por um ruído aleatório e a probabilidade de contaminação de cada estado se dá por uma probabilidade pequena e fixada. No Modelo Zero Inflado definido por Garcia e Moreira (2015), o processo contaminado é dado pelo produto entre o processo original e uma variável aleatória binomial independente que assume os valores zero ou um. Nesse modelo, não é possível que todos os estados do processo original sejam contaminados. Um outro modelo de Contaminação Estocástica chamado de Contaminação por Congruência foi introduzido por Collet, Galves e Leonardi (2008). Diferentemente do Modelo Zero

Inflado, no Modelo de Contaminação por Congruência todos os estados do processo original podem ser contaminados. Nele, o processo contaminado se dá pela soma do processo original a uma variável aleatória binomial independente do processo que toma valores zero ou um com uma probabilidade pequena fixada. Um outro modelo a ser considerado foi introduzido por Garcia e Moreira (2015). A ideia é que, dada duas cadeias de ordem variável tomando valores no mesmo alfabeto finito, a cada instante de tempo o processo contaminado escolhe um dos dois processos originais com probabilidade fixa.

Neste trabalho, verificamos o bom desempenho do Algoritmo BIC proposta por Csiszar e Talata (2005) quando utilizamos amostras contaminadas, desta maneira viabilizando a aplicação em modelos de umidade relativa do ar considerados nesse trabalho. Para a aplicação do Cadeias de Ordem Variável, modelamos dados de umidade relativa do ar de Brasília considerando o período de 1962 à 2018 e tentamos prever a possibilidade do próximo dia ter a umidade adequada pra saúde, estar em estado de atenção, estado de alerta ou estado de emergência. Em um segundo momento, dividimos a série histórica em dois períodos verificamos se houveram mudanças na umidade relativa do ar em Brasília devido ao processo de urbanização ao longo do tempo. Todas as simulações e estimativas foram realizadas através do ambiente RStudio Versão 3.5.0.

1 Revisão Bibliográfica

Neste capítulo, começaremos introduzindo os conceitos básicos de uma Cadeia de Ordem Variável. Nas Seção 1.5 será definido formalmente o Modelo de Contaminação Estocástica.

1.1 Definições e Notações Básicas

Considere o alfabeto $\mathcal{A} = \{0, 1, 2, 3, ..., N-1\}$ com cardinalidade $|\mathcal{A}| = N$. Dados dois inteiros n e m, define-se $a_m^n = a_m, a_{m+1}, a_{m+2}..., a_n$ como uma sequência de símbolos do alfabeto $\mathcal{A}$, onde $\ell(a_m^n) = n - m + 1$ é o comprimento dessa sequência. Percebam que os índices vão de m até n, de tal forma que $m < n$. Caso $m > n$ denota-se $\ell(a_m^n) = \emptyset$ uma sequência vazia. Dado o alfabeto $\mathcal{A}$, denota-se $\mathcal{A}^k$ como o conjunto de todas as sequências de tamanho k de $\mathcal{A}$ e $\mathcal{A}^0$ o conjunto de todas as sequências vazias. É importante também definir

- $\mathcal{A}_{-\infty}^{-1} = \mathcal{A}^{\{...,-2,-1\}}$, o conjunto de todas as sequências semi-finitas de $\mathcal{A}$,
- $\mathcal{A}^* = \bigcup_{i=0}^{\infty} \mathcal{A}_{-i}^{-1}$, o conjunto de todas as sequências finitas de $\mathcal{A}$.

Dado duas seqûencias finitas a e b , representa-se como ab a concatenação da seqûencia com comprimento $\ell(a) + \ell(b)$. A título de exemplo, tomemos $a_1^m = a_1, a_2, ..., a_m$ e $b_1^n = b_1, b_2, ..., b_n$. Portanto ab é igual $a_1, a_2, ...a_m, b_1, ...b_{n-1}, b_n$. Caso alguma das duas seja uma seqûencias vazia, por exemplo $b = \oslash$, então a concatenação ab é igual a $a_1, a_2, ..., a_m$ que é o próprio a. É dito $s \in \mathcal{A}^*$ um sufixo de a, se existir alguma sequência $b \in \mathcal{A}^* \cup \mathcal{A}_{-\infty}^{-1}$ tal que a possa ser escrito como a concatenação $a = bs$. Denotamos então $suf(a)$ o maior sufixo de a.

Definição 1. *Um conjunto $\mathcal{T} \in \mathcal{A}^* \cup \mathcal{A}_{-\infty}^{-1}$ de seqûencias pode ser escrita como uma árvore probabilística se nenhuma dessas seqûencias pertencentes a $\mathcal{T}$ for sufixo de outra seqûencia pertencente a $\mathcal{T}$. Chamamos essa propriedade de propriedade do sufixo. Os elementos da árvore $\mathcal{T}$ são chamados de folhas de $\mathcal{T}$ e um nó interno é um sufixo de uma folha.*

Uma árvore $\mathcal{T}$ é dita completa se cada nó interno tem $|\mathcal{A}|$ descendentes, onde os descendentes de um nó interno são todas as sequências $bs, a \in \mathcal{A}$. Se nenhuma sequência $s \in \mathcal{T}$ puder ser substituída por um sufixo de s sem violar a *propriedade do sufixo*, então dizemos que a árvore é irredutível. Essa noção foi introduzida em Csiszár e Talata (2006) e generaliza o conceito de árvore completa. Denotamos por $|\mathcal{T}|$ a cardinalidade de $\mathcal{T}$. A profundidade de uma árvore $\mathcal{T}$ é definida como

$$h(\mathcal{T}) = \max\{l(s), s \in \mathcal{T}\}.$$

Se $h(\mathcal{T}) < \infty$, dizemos que $\mathcal{T}$ é *limitada*. Caso contrário, dizemos que $\mathcal{T}$ é *ilimitada*. Dado um inteiro K, $\mathcal{T}|_K$ denota a árvore $\mathcal{T}$ truncada em K, ou seja,

$$\mathcal{T}|_K = \{s \in \mathcal{T} : l(s) \leq K\} \bigcup \{s \in \mathcal{A}^k : s \prec s' \text{ para algum } s' \in \mathcal{T}\}. \tag{1.1}$$

Nesse trabalho consideraremos o processo $X = \{X_t, t \in \mathbb{Z}\}$ estacionário e ergótico sobre o alfabeto $\mathcal{A} = \{0, 1, ..., N-1\}$. Assumimos que o processo X é compatível com a probabilidade de transição $p_X(.|.)$. Então:

$$p_X(a|\omega) = \mathbb{P}(X_0 = a | X_{-1} = \omega_{-1}, X_{-2} = \omega_{-2}, ...). \tag{1.2}$$

para todo $\omega \in \mathcal{A}_{-\infty}^{-1}$ e para todo $a \in \mathcal{A}$. Para $\omega \in \mathcal{A}_{-j}^{-1}$ a probabilidade estacionária do cilindro definida por essa sequência é definida como

$$\mu_X(\omega) = \mathbb{P}(X_{-j}^{-1} = \omega) \tag{1.3}$$

Pela hipótese de estacionariedade, a distribuição de probabilidade $\mathbb{P}$ de uma cadeia de ordem variável é completamente especificada por suas probabilidades de transição $\mathbb{P}\{X_0 = x_0 | t(x_{-\infty}^{-1})\}$. Desta forma, uma maneira apropriada de representar o espaço de estados é através de uma árvore probabilística (árvores de contexto). Ressaltamos que nenhuma sequência pode ser sufixo de outra sequência. Sendo assim, se $(x_l, ...x_1)$ é um contexto, então nenhuma das sequências $(x_j, ..., x_1)$ com $i \leq j \leq l-1$, é um contexto. Portanto, representamos a árvore de contextos da seguinte maneira:

1. O primeiro nó é a raiz;
2. Os ramos são os passados relevantes e crescem de cima para baixo;
3. Cada nó tem no máximo $|\mathcal{A}|$ descendentes;
4. Os contextos são os ramos que ligam o último nó à raiz;
5. Cada contexto é representado por um ramo completo;
6. O contexto $l = t(x_{-\infty}^{-1})$ é representado por um ramo, cujo sub-ramo do topo é determinado por x_{-1}, o próximo sub-ramo é determinado x_{-2} e assim sucessivamente.

Exemplo 1 (Árvore de Contextos)**.** *Considere o alfabeto* $A = \{0, 1, 2\}$*. O conjunto de sequências* $\{00, 10, 20, 01, 21, 2\}$ *que vão da base ao topo são os contextos e satisfazem a propriedade do sufixo. Portanto é de fato uma árvore. Representamos graficamente a árvore* $\mathcal{T} = \{00, 10, 20, 01, 21, 2\}$ *abaixo:*

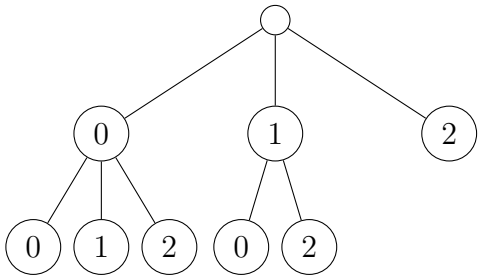


Figura 1 – Exemplo da Árvore de Contextos

Como os nós 1 e 2 não possuem todos os descendentes, dizemos que a árvore $\mathcal{T}$ não é completa. Percebe-se ainda que o maior caminho da base ao topo, o maior contexto, tem tamanho 2. Dizemos então que a árvore tem profundidade 2.

Com o objetivo de estimarmos a árvore de contexto de um processo X, primeiramente precisamos considerar que X satisfaz as seguintes definições:

Definição 2. *Um processo X é não nulo se*

$$\alpha_x = inf\{p_x(a|w) : a \in \mathcal{A}, w \in \mathcal{A}\} > 0. \tag{1.4}$$

Definição 3. *Dizemos que um processo X tem taxa de continuidade somável quando*

$$\beta_x = \sum_k \beta_{k,x} < \infty, \tag{1.5}$$

onde β_x é definido da seguinte maneira:

$$\beta_x = sup\{|1 - \frac{p_x(a|w)}{p_x(a|v)}| : a \in \mathcal{A}, v, w \in \mathcal{A}_{-\infty}^{-1} \ com \ \ w_{-k}^{-1} = v_{-k}^{-1}\}. \tag{1.6}$$

A taxa de continuidade do processo é uma propriedade desejada visto que esperamos que dois passados que coincidem nos últimos k símbolos tenham o mesmo peso na predição do próximo símbolo da sequência. Note que a condição de não-nulidade do processo é necessária para que se possa definir a taxa de continuidade do processo. A seguir, definiremos de maneira formal um contexto.

Definição 4. *Uma sequência $s \in \mathcal{A}^k$ é um contexto finito para o processo X se satisfaz*

1. $\mu_X(s) > 0$;
2. Para toda sequência semi-infinita $x_{-\infty}^{-1}$ que tem s como sufixo

$$\mathbb{P}(X_0 = a|X_{-\infty}^{-1} = x_{-\infty}^{-1}) = p_X(a|s) \text{ para todo } a \in \mathcal{A}; \tag{1.7}$$

3. Nenhum sufixo de s satisfaz o item anterior.

Um contexto infinito é uma sequência de símbolos semi-infinita $\omega = x_{-\infty}^{-1}$ cujos sufixos x_{-k}^{-1}, com $k = 1, 2, ...$ tem probabilidade positiva e nenhum deles é um contexto.

O conjunto de todos os contextos de um processo X formam a árvore de contextos de X que será uma árvore irredutível. A seguir, definiremos o conceito de árvore probabilística de contextos e quando um processo X é compatível com esse modelo.

Definição 5. *Uma árvore probabilística de contextos em $\mathcal{A}$ é um par ordenado $(\mathcal{T}, \bar{p})$ que satisfaz:*

1. *$\mathcal{T}$ é uma árvore irredutível;*

2. *$\bar{p} = \{\bar{p}(.|s), s \in \mathcal{T}\}$ é uma família de probabilidades de transição sobre $\mathcal{A}$.*

Definição 6. *Dizemos que o processo X é compatível com a árvore probabilística de contextos $(\mathcal{T}, \bar{p})$ se satisfaz*

1. *$\mathcal{T}$ é a árvore de contextos do processo X*

2. *Para qualquer $\omega \in \mathcal{T}$ e $a \in \mathcal{A}$, $p_X(a|\omega) = \hat{p}(a|w)$*

Denotamos por $\mathcal{T}_X$ a árvore de contextos de X. Se $\mathcal{T}_X$ tem profundidade $h(\mathcal{T}_X) = k < \infty$, então o processo X é uma cadeia de Markov de ordem k. As árvores probabilísticas de contexto são comumente utilizados para descrever processos estocásticos uma vez que são mais flexíveis e econômicos no número de parâmetros do que um modelo de Markov, possuindo $(|\mathcal{A}| - 1)|\mathcal{T}_X|$ parâmetros em vez de $(|\mathcal{A}| - 1)|\mathcal{A}|^k$ parâmetros necessários para estimar uma cadeia de Markov de ordem k.

Definição 7. *Considere uma amostra do processo $X = \{X_1, X_2, X_3...\}$ e considere w_1^j um vetor tal que $w_1^j \in \bigcup_{m=1}^{n} A^m$. Então temos*

$$N_n(w) = \sum_{t=0}^{n-\ell(w)} 1_{Z_{t+1}^{t+\ell(w)}=w} \qquad e \qquad \hat{P}(w) = \frac{N_n(w)}{n}, \tag{1.8}$$

que representam respectivamente número de vezes que a sequência w aparece em um amostra de tamanho n e a probabilidade de ocorrência. Podemos notar que $N_n(w) = \sum_{a\in\mathcal{A}} N_n(s, a)$. Se $s = \emptyset$, dizemos que $N_n(s, a) = \sum_{i=d+1}^{n} \mathbb{I}(X_{i-l(s)}^{i-1} = s$ e $N_n(s) = n - d$. O conjunto de todas as sequências $s \in \bigcup_{j=0}^{d} \mathcal{A}^j$ que aparecem na amostra um número de vezes $d \geq 1$ é descrito por $\mathcal{V}_n$, onde

$$\mathcal{V}_n = \{s \in \bigcup_{j=0}^{d} \mathcal{A}^j : N_n(s) \geq 1\}. \tag{1.9}$$

O estimador de probabilidade de transição será dado por:

$$\hat{p}_z(a|w)_n = \frac{N_n(wa)+1}{N_n(w.)+|A|}. \tag{1.10}$$

Note que a definição de $\hat{p}_z(a|w)_n$ é interessante pois ela equivale assintoticamente ao Estimador de máxima Verossimilhança e evitamos assim uma definição adicional para quando $N(w.)=0$. A seguir, definiremos o que é uma árvore factível. Essa definição é importante pois queremos selecionar as possíveis árvores de contexto do processo X. Uma árvore $\mathcal{T}$ vai ser factível se:

1. $s \in \mathcal{V}_n$ para todo $s \in \mathcal{T}$;

2. Cada uma das sequências $s' \in \mathcal{V}_n$ é tal que $s' \preceq s$ ou $s \prec s'$ para algum $s \in \mathcal{T}$.

O conjunto de todas as árvores factíveis será denotado por $\mathcal{F}_n$. A ideia é estimar a árvore de contextos do processo $\mathcal{T}_X$ a partir de uma amostra de X. Para isso, precisamos escolher uma árvore factível que se aproxime de $\mathcal{T}_X$. Se $h(\mathcal{T}_X) < \infty$, então devemos escolher o inteiro d de modo que $h(\mathcal{T}_X) \leq d$ para que exista uma árvore factível que coincida com $\mathcal{T}_X$. Para estimar $\mathcal{T}_X$, não é necessário conhecer precedentemente sua profundidade, então d pode ser uma função crescente de n.

Exemplo 2 (Estimação das Probabilidades de Transição)**.** *Considere X uma Cadeia de Ordem Variável tomando valores num alfabeto $\mathcal{A} = \{0,1\}$ e com a árvore de contextos $\mathcal{T}_X = \{0, 01, 11\}$. Seja 111001011010111 uma amostra aleatória do processo X de tamanho 15.*

Observe que a profundidade da árvore é $d(\mathcal{T}_\mathcal{X}) = 2$, visto que, os contextos de maior tamanho são 01 e 11. Note que o número de ocorrências dos contextos $\omega = 0$, $v = 01$ e $u = 11$ foram dadas, respectivamente por $N_{15}(\omega) = 5$, $N_{15}(v) = 4$ e $N_{15}(u) = 5$. Com o objetivo de estimar as probabilidades de transição, é necessário determinar o número de ocorrências da concatenação entre cada contexto com estado 0. Foram obtidas $N_{15}(00) = 1$, $N_{15}(010) = 2$ e $N_{15}(110) = 2$. As probabilidades de transição estimadas foram $\hat{p}_X(0|0)_{15} = 0{,}286$, $\hat{p}_X(0|01)_{15} = 0{,}5$ e $\hat{p}_X(0|11)_{15} = 0{,}428$.

1.2 O Algoritmo Contexto

O Algoritmo contexto proposto por Rissanen (1983) se baseia na diferença entre as probabilidades empíricas de transição do maior sufixo de um contexto e seus descendentes para a poda da árvore. Caso essa diferença seja menor do que um certo valor o contexto é podado.

Definição 8. *A divergência de Kullback-Leibler, definida para medidas de probabilidade P e Q em $\mathcal{A}$, é dada por*

$$D(P;Q) = \sum_{a\in\mathcal{A}} P(a) log \frac{P(a)}{Q(a)}. \tag{1.11}$$

Dada uma sequência $s \in \mathcal{V}_n$, então

$$\Lambda_n(s) = \sum_{b\in\mathcal{A}:bs\in\mathcal{V}_n} N_n(bs) D(\hat{p}_n(.|bs); \hat{p}_n(.|s)). \tag{1.12}$$

O valor limite utilizado no Algoritmo Contexto é denotado por δ_n, onde $(\delta_n)_{n\in\mathbb{N}}$ representa uma sequência de números reais tal que $\delta_n \to \infty$ e $\delta_n/n \to 0$ quando $n \to \infty$. Podemos descrever o funcionamento do Algoritmo Contexto da seguinte maneira: Primeiramente, uma árvore maximal é produzida de tal forma que sua construção considera todos os ramos que possuem um comprimento pré estabelecidos e que aparecem um número mínimo de vezes na amostra. Em seguida, o algoritmo $\Lambda_n(s)$ poda a árvore de baixo para cima usando para isso o critério de decisão de poda onde o valor limiar é δ_n. Esse procedimento de poda é efetuado em cada contexto testado até que se tenha uma árvore irredutível.

Para uma amostra X_1^n , podemos também obter $\hat{\mathcal{T}}_C(X_1^n)$ a partir da função $C_s(X_1^n)$, definida para todo $s \in \mathcal{V}_n$ e dada por

$$C_s(X_1^n) = \begin{cases} 0, & \text{se } N_n(s) \leq 1 \text{ ou } l(s) = d, \\ max\{\mathbb{I}(\Lambda_n(s) \geq \delta_n),\ max_{b\in\mathcal{A}} C_{bs}(X_1^n)\}, & \text{se } N_n(s) > 1 \text{ e } l(s) < d. \end{cases}$$

Definição 9. *O estimador $\hat{\mathcal{T}}_C(X_1^n)$ da árvore de contextos de X é o conjunto dado por*

$$\hat{\mathcal{T}}_C(X_1^n) = \{s \in \mathcal{V}_n : C_s(X_1^n) = 0 \ e \ C_{s'}(X_1^n) = 1 \ \text{para todos} s' \prec s\}. \tag{1.13}$$

1.3 Uma versão Modificada do Algoritmo Contexto

Uma versão modificada do Algoritmo Contexto de Rissanen foi introduzido por Galves e Leonardi em 2008. O estimador da árvore de contexto computa a distância entre as probabilidades empíricas para uma sequência w e uma sequência $suf(w)$. O operador é definido da seguinte maneira:

$$\Delta_n(w) = max|\hat{p}_z(a|w)_n - \hat{p}_z(a|suf(w))_n|. \tag{1.14}$$

Definição 10. *Para todo $\delta > 0$ e $d < n$, o estimador da árvore de contextos $T_n^{\delta,d}$ é o conjunto de todas as sequências $s \in \bigcup_{k=1}^{d-\ell(s)} A^k$ de tal maneira que $\Delta_n(asuf(f)) > \delta$ para algum $a \in A$ e $\Delta_n(us) \geq \delta$, para todo $u \in \bigcup_{k=1}^{d-\ell(s)}$.*

O algoritmo de estimação trunca as sequências w que não satisfazem o critério de poda,considerando então o $suf(w)$ como um novo candidato a contexto. Perceba que as contantes $\delta > 0$ e $d < n$ são fundamentais para o estimador uma vez que inicialmente é considerada a árvore de contexto maximal.

1.4 O critério BIC

Seja $X_1, X_2, ..., X_n$ uma amostra do processo X. A seleção de uma árvore factível $\mathcal{T}_0 \subset \mathcal{F}_n$ que estime $\mathcal{T}_X$ deve considerar alguns aspectos: A função de verossimilhança da amostra e a complexidade da árvore. O intuito é escolher $\mathcal{T}_0$ de modo que a função de verossimilhança da amostra seja comparativamente alta, com preferência por modelos menos complexos. Definimos o critério de informação Bayesiana (BIC) a seguir.

Definição 11. *Considerando uma amostra do processo X, o Critério de informação Bayesiana (BIC) para uma árvore factível é dado por*

$$BIC_t(X_1^n) = -logML_t(X_1^n) + c|T|logn \tag{1.15}$$

Ressaltamos que no critério BIC a penalização não é constante, ela varia de acordo com o tamanho da amostra. Em Czizár e Talata (2006) foi escolhido a constante $c = (|A| - 1)/2$ para a profundidade da árvore estimada.O estimador BIC é dado por

$$\hat{T}_{BIC}(X_1^n) = arg\ \ min\ \ BIC_t(X_1^n) \tag{1.16}$$

1.5 Modelos de Contaminação Estocástica

É assumido que o processo original está contaminado por algum ruído aleatório com um parâmetro de pertubação fixado. Consideramos o Modelo de Contaminação Estocástica Zero Inflado, definido em Garcia e Moreira (2015). Para o modelo, consideramos x como um processo estocástico não nulo e com taxa de continuidade somável, uma vez que esperamos que dois passados coincidindo nos últimos k símbolos tenham a mesma influência na predição do próximo símbolo da sequência, a medida que k cresce. Denotamos por $Z = \{Z_t, t \in \mathbb{Z}\}$ o processo estocasticamente perturbado.

Definição 12. *Considere um processo $X = \{X_t, t \in \mathbb{Z}\}$ estacionário e ergódico tomando valores no alfabeto $\mathcal{A} = \{0, 1\}$. Seja $\xi = \{\xi_t : t \in \mathbb{Z}\}$ uma sequência de variáveis aleatórias i.i.d. com distribuição Bernoulli, independente do processo X, tal que*

$$\mathbb{P}(\xi_t = 1) = 1 - \varepsilon,$$

onde ε é o parâmetro de pertubação fixado no intervalo $(0, 1)$. O Modelo Zero Inflado é dado por

$$Z_t = X_t.\xi_t, t \in \mathbb{Z}. \tag{1.17}$$

Como ε representa a probabilidade de $\xi_t = 0$, quanto maior for ε maior a probabilidade de $Z_t = 0$ uma vez que Z_t é dado pelo produto de $X_t.\xi_t$. Pode-se observar então que no modelo Zero Inflado a perturbação pode ocorrer apenas quando $X_t = 1$ e $\xi_t = 0$.

2 Simulações de Cadeias de Ordem Variável

Neste capítulo avaliamos o desempenho dos estimadores BIC e Contexto Modificado apresentados nas Definições 10 e 11 para amostras contaminadas segundo o modelo de Contaminação Zero Inflado. Antes de partirmos para os modelos de contaminação, é necessário um estudo prévio com amostras não contaminadas para procurar o valor mais adequado da constante de penalização δ. Queremos verificar o comportamento do estimador a medida que variamos a constante de penalização δ a fim de melhorar a precisão do mesmo. Em um segundo momento, utilizando o valor mais adequado δ,queremos testar a robustez do estimador para diferentes tamanhos de amostras e proporções de contaminação ε.

2.1 Cadeias de Ordem Variável não Contaminadas

Nesta seção, verificamos a eficiência dos estimadores das árvores de contextos apresentado nas Definições 10 e 11, através de simulações com amostras não contaminadas. Consideramos Cadeias de Ordem variável distintas tomando valores no alfabeto $\mathcal{A} = \{0, 1, 2, 3\}$ com profundidade igual a três. Essas cadeias são representadas pelas árvores de contextos das Figuras 2, 3 e 4.

Com o objetivo de estudarmos a eficiência dos algoritmos, utilizamos diferentes tamanhos de amostras com diferentes constantes δ. Primeiramente, foram construídas três árvores distintas que representam três Cadeias de Ordem Variável diferentes. A ideia é testar os estimadores em cenários diferente para que assim possamos entender melhor o seu comportamento.

Com base em cada uma das três árvores, geramos 100 amostras distintas com os tamanhos 1000, 5000, 10000, 20000 e 50000. Em seguida, utilizando os estimadores das árvores de contextos, verificamos o percentual de retornos corretos em cada um dos cenários, para cada constante δ e para cada tamanho de amostra.

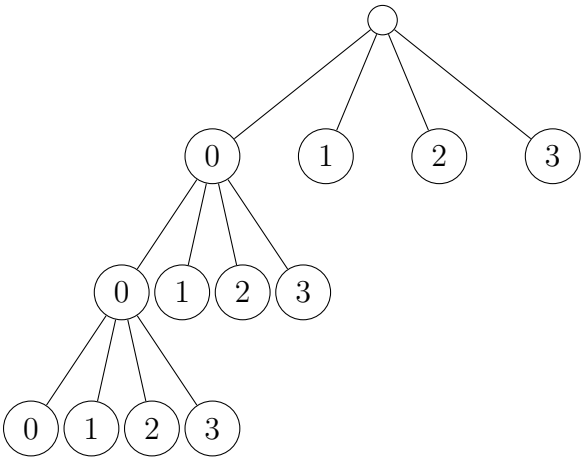


Figura 2 – Árvore de Contextos 1

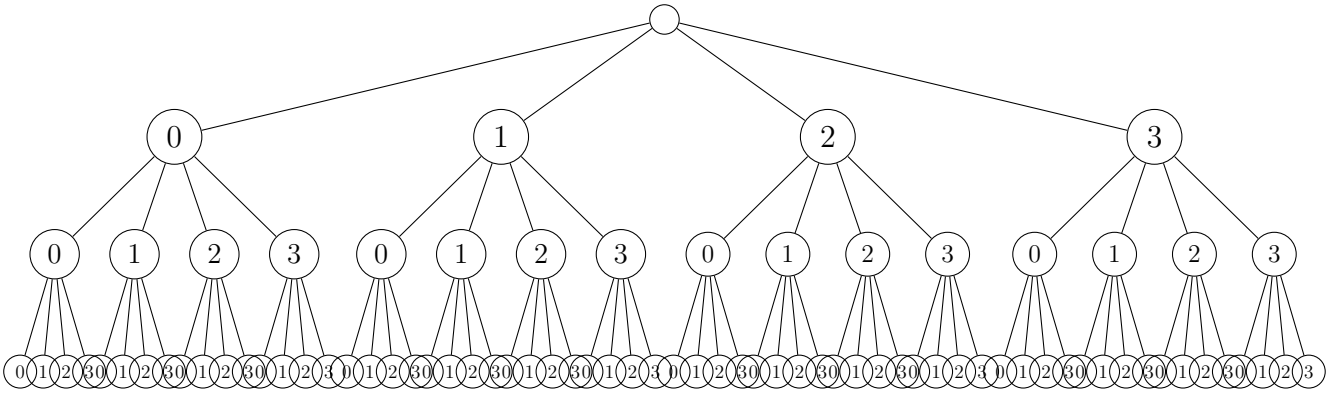


Figura 3 – Árvore de Contextos 2

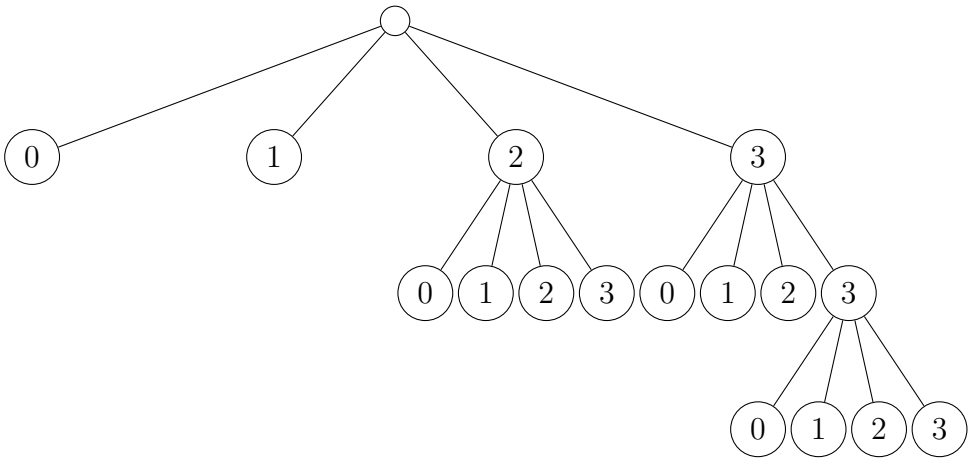


Figura 4 – Árvore de Contextos 3

Para a Árvore 1, observamos que a medida que variamos a constante de poda δ temos uma sensibilidade maior para valores próximos de 0.13 nos dois algoritmos. Em todos os tamanhos de amostras simulados, esse valor de δ apresentou uma maior proporção de acertos. Para as amostras de tamanho 1000, a proporção de acertos chegou aos 0,28, enquanto para os demais tamanhos de amostras, alcançou valores próximos a 1. É interessante observar que conforme o tamanho da amostra aumenta, ganhamos mais liberdade na escolha do δ no algoritmo BIC. Para a maior amostra, com n=50000, qualquer valor de delta entre 0.13 e 0.48 é igualmente eficiente na estimação da árvore de contextos.

Observamos na Figura 6 que o Contexto Modificado é mais sensível a variações na contante de poda. Mesmo com um tamanho maior da amostra, o maior percentual de acertos ocorre em torno da constante de poda 0.13.

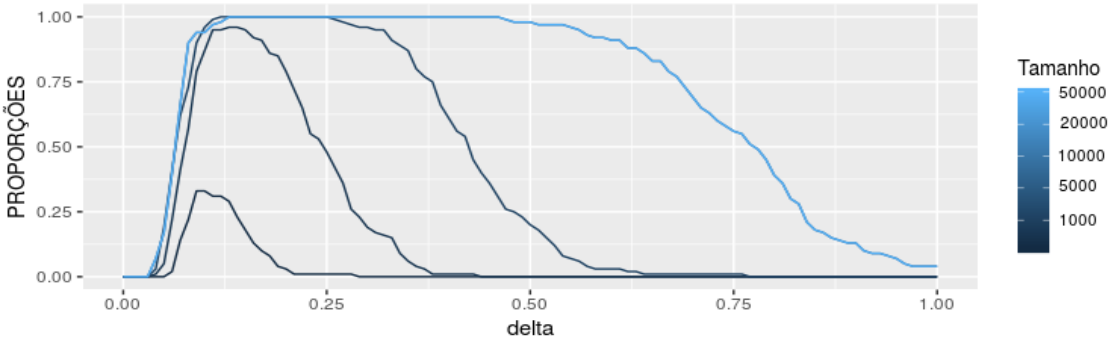


Figura 5 – Proporções de acerto para Árvore 1 - BIC

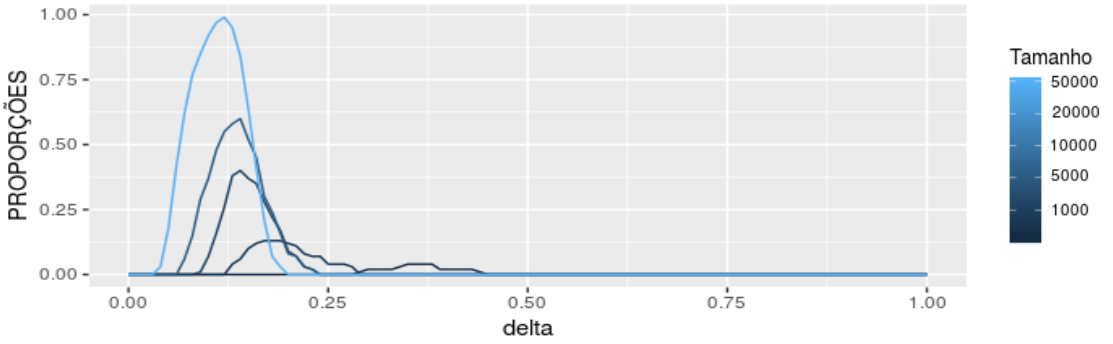


Figura 6 – Proporções de acerto para Árvore 1 - Contexto Modificado

Para a Árvore 2, observamos que os estimadores são mais eficientes para pequenos valores de δ. Isso pode ser explicado pelo método de poda dos algoritmos que inicialmente consideram uma árvore completa e, em seguida, a podam de acordo com o limiar δ. A ideia é que, quanto menor a constante δ, menos será podado. Como a Árvore 2 é completa, esperamos que o estimador não pode nenhum galho, por isso a maior eficiência para valores de δ próximos de 0.

Podemos observar na Figura 7 o comportamento do estimador BIC com tamanhos distintos de amostras. Repare que o decaimento é constante porém, conforme vimos para

Árvore 1, quanto maior o tamanho da amostra maior é a liberdade que temos para a escolha da constante de poda. Para o Contexto Modificado, a amplitude de escolha da constante é menor e conforme a amostra cresce a função decai mais rapidamente.

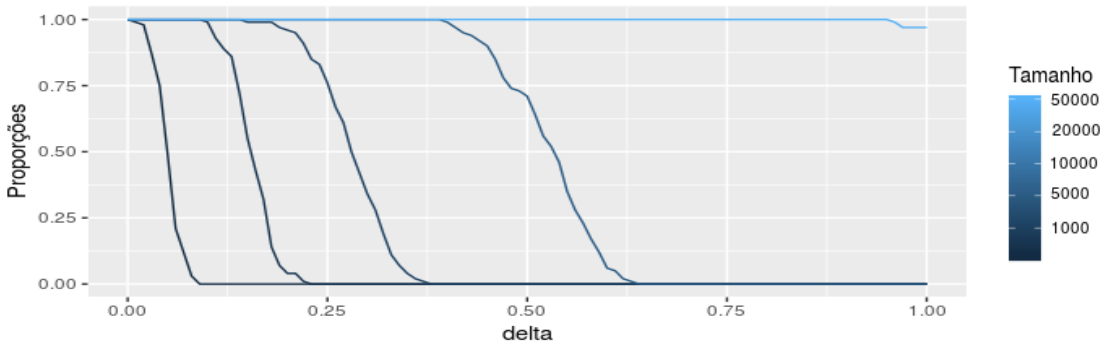


Figura 7 – Proporções de acerto para Árvore 2 - BIC

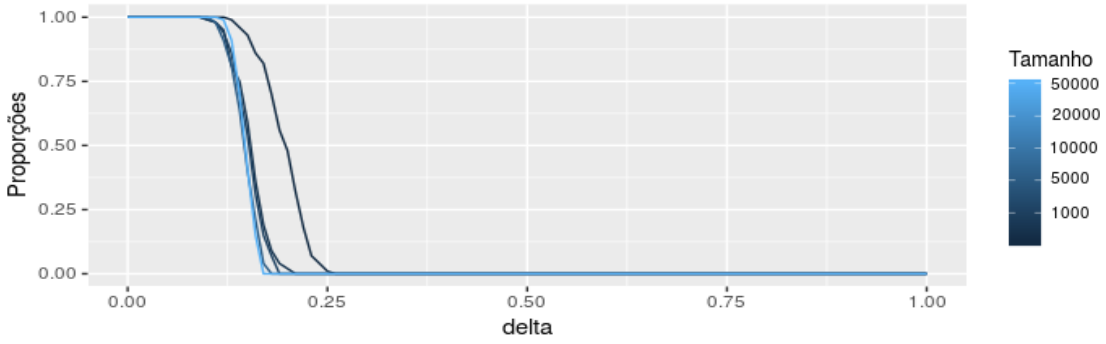


Figura 8 – Proporções de acerto para Árvore 2 - Contexto Modificado

Para a Árvore 3, observamos um comportamento semelhante a Árvore 1. Como a Árvore 3 não é completa, para valores muito grandes de δ o estimador poda muito, enquanto para valores muito pequenos o estimador poda pouco. Notamos então que o estimador é mais eficaz a medida que o limiar δ se aproxima da constante 0.13.

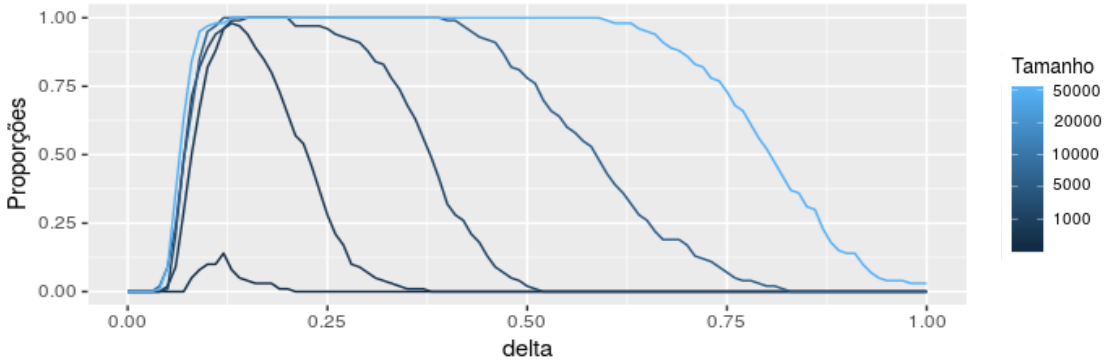


Figura 9 – Proporções de acerto para Árvore 3 - BIC

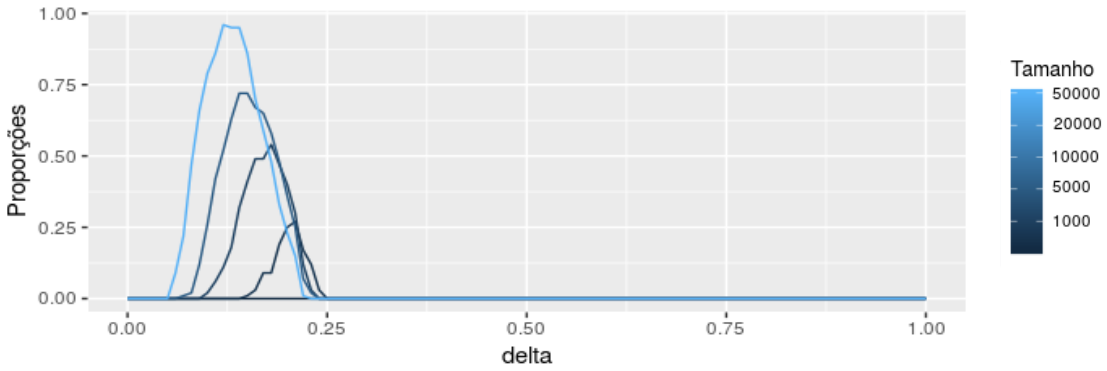


Figura 10 – Proporções de acerto para Árvore 3Contexto Modificado

2.2 Cadeias de Ordem Variável Contaminadas

Nesta seção, introduzimos contaminação nos processos referentes as Árvores 1, 2 e 3 segundo o modelo de contaminação Zero Inflado. A fim de estimar os processos por meio das amostras contaminadas, utilizamos o parâmetro $\delta = 0.13$, por ter sido o mais eficiente nas simulações não contaminadas. Realizamos 100 simulações com amostras de tamanho 5000 e 30000 variando o parâmetro de pertubação ε. Após as simulações, avaliamos o percentual de retornos corretos para cada árvore. Os resultados obtidos foram apresentados nas tabelas a seguir.

Tabela 1 – Proporção de retornos, segundo o modelo Zero Inflado, n=5.000 BIC

Árvores de Contexto	**Parâmetro de Pertubação**				
	0.01	**0.05**	**0.10**	**0.15**	**0.20**
Árvore 1	0.95	0.68	0.41	0.11	0.07
Árvore 2	0.96	0.87	0.71	0.58	0.36
Árvore 3	0.93	0.83	0.62	0.48	0.25

Tabela 2 – Proporção de retornos, segundo o modelo Zero Inflado, n=5.000 Contexto Modificado

Árvores de Contexto	**Parâmetro de Pertubação**				
	0.01	**0.05**	**0.10**	**0.15**	**0.20**
Árvore 1	0.35	0.30	0.00	0.00	0.00
Árvore 2	1.00	1.00	1.00	1.00	0.98
Árvore 3	0.23	0.13	0.06	0.05	0.01

As Tabelas 1 e 2 apresentam os resultados dos acertos para cada árvore segundo o modelo de contaminação Zero Inflado. Nesse primeiro momento, consideramos uma amostra com n=5000 pois queríamos testar a eficiência do estimador para uma amostra menor. Note que para o BIC as três árvores não atingem 100% de acerto nesse tamanho de amostra. Isso se deve a grande diferença entre o número de parâmetros da árvore original e o número de parâmetros a serem estimados . São muitos parâmetros a serem podados para uma amostra de tamanho 5000 e o estimador tem dificuldade para estimar corretamente a árvore de contextos conforme o nível de contaminação aumenta. Note que a Árvore 2 é a que possui menos contextos a serem podados em relação a árvore completa, por isso os dois estimadores se desempenharam melhor em relação as outras duas árvores mas com bastante diferença entre os dois. Essa característica pode ser explicada pelo caráter do estimador Contexto Modificado que costuma gerar árvores com mais contextos e por isso se desempenha melhor do que o algoritmo BIC em árvores completas.

Aumentando o tamanho da amostra para 30000 e considerando o modelo de contaminação Zero Inflado, podemos observar os resultados obtidos através das Tabelas 3 e 4. Para uma amostra maior, observamos que o estimador se saiu melhor para as três árvores em relação as amostras de tamanho 5000.

Tabela 3 – Proporção de retornos, segundo o modelo Zero Inflado, n=30.000 BIC

Árvores de Contexto	**Parâmetro de Pertubação**				
	0.01	**0.05**	**0.10**	**0.15**	**0.20**
Árvore 1	1.00	1.00	1.00	1.00	0.96
Árvore 2	1.00	1.00	1.00	1.00	1.00
Árvore 3	1.00	0.92	0.41	0.05	0.00

Tabela 4 – Proporção de retornos, segundo o modelo Zero Inflado, n=30.000 Contexto Modificado

Árvores de Contexto	**Parâmetro de Pertubação**				
	0.01	**0.05**	**0.10**	**0.15**	**0.20**
Árvore 1	0.81	0.15	0.00	0.00	0.00
Árvore 2	1.00	1.00	1.00	1.00	0.86
Árvore 3	0.94	0.85	0.74	0.71	0.46

Para as simulações das amostras contaminadas das Árvores 1, 2 e 3, observamos que para um baixo nível de contaminação, ε=0.01, o estimador BIC recuperou corretamente 100% das árvores. Conforme nível de contaminação aumenta vemos que, assim como nas Tabelas 1 e 2, os percentuais de acerto diminuem. Porém, ao compararmos as duas tabelas, vemos que o decaimento dos percentuais de acerto é mais lento na amostra n=30000, o que demonstra o melhor desempenho dos estimadores em amostras maiores.

Podemos ver que mesmo as Árvores 1 e 3 sendo incompletas e tendo a mesma profundidade, os estimadores se desempenham de maneira distinta nos dois casos. Para a Árvore 3 tivemos um comportamento peculiar para o estimador BIC, onde a amostra de tamanho 5000 desempenha melhor para maiores níveis de contaminação do que a amostra de 30000. Porém, o estimador se desempenha melhor com baixos níveis de contaminação em relação a amostra menor. As Figuras a seguir apresentam a comparação de cada árvore para os dois tamanhos de amostras utilizados.

Figura 11 – Proporções de acerto para Árvore 1, BIC

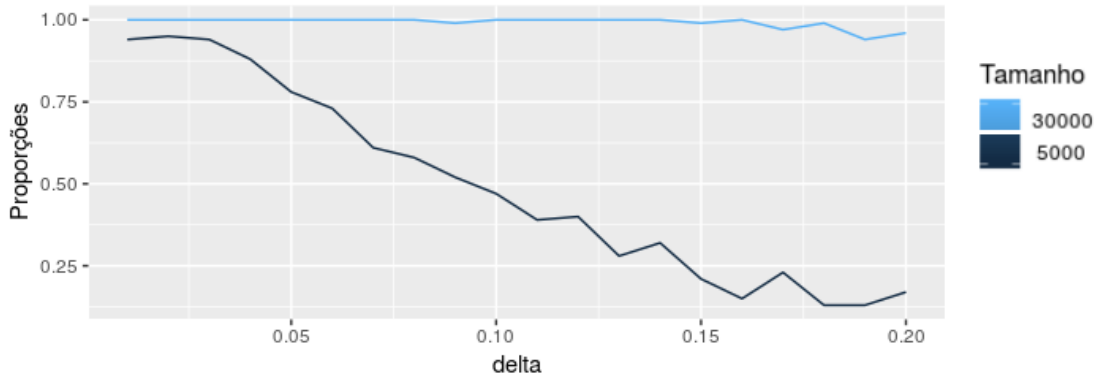


Figura 12 – Proporções de acerto para Árvore 2, BIC

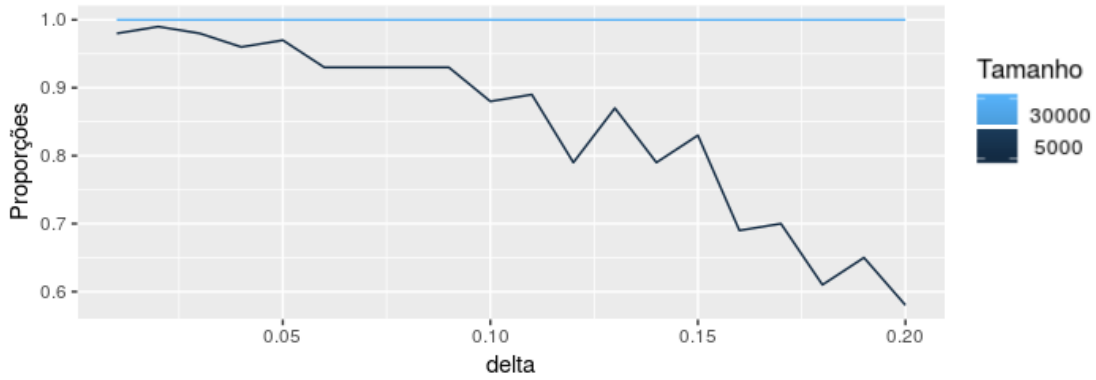


Figura 13 – Proporções de acerto para Árvore 3, BIC

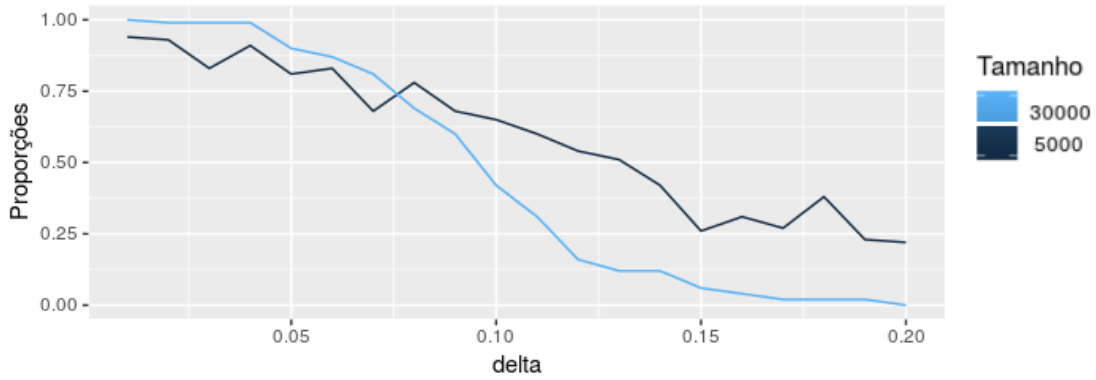


Figura 14 – Proporções de acerto para Árvore 1, Contexto Modificado

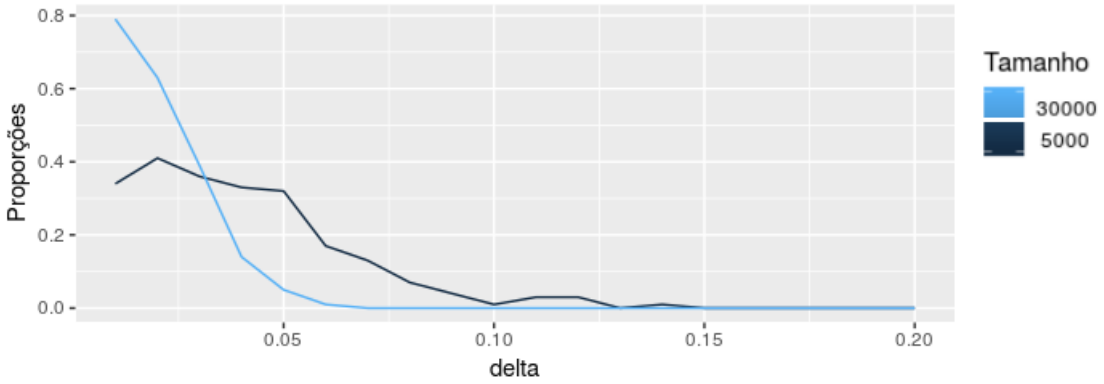


Figura 15 – Proporções de acerto para Árvore 2, Contexto Modificado

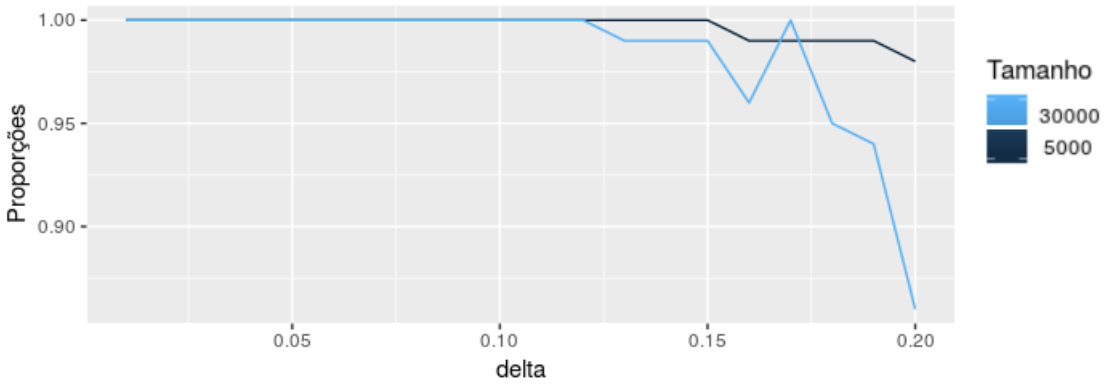
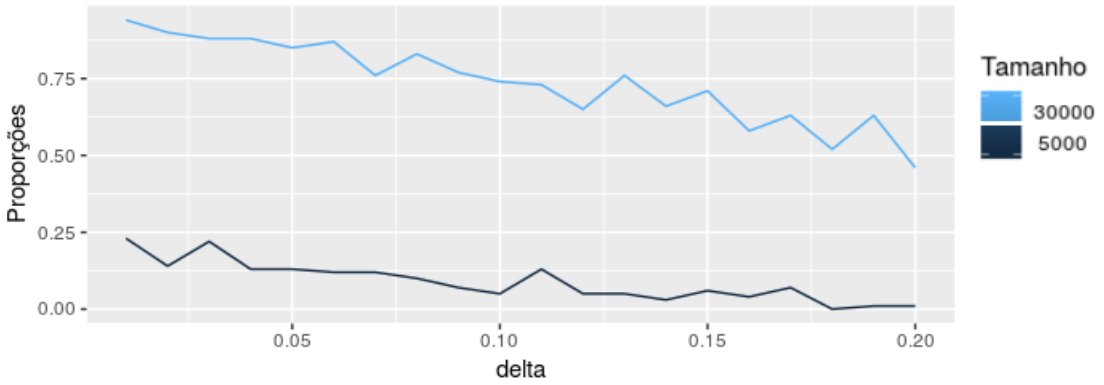


Figura 16 – Proporções de acerto para Árvore 3, Contexto Modificado



3 Aplicação de Cadeias de Ordem Variável a dados Meteorológicos

O estudo do clima resulta da combinação de vários fatores geográficos (sol, topografia, corpos d'agua, ventos, vegetação, etc) e de elementos como a temperatura, umidade relativa do ar, precipitação e evaporação. Para o estudo do clima urbano, é necessário também considerar o possível impacto que o processo de urbanização pode causar nesses elementos climáticos.

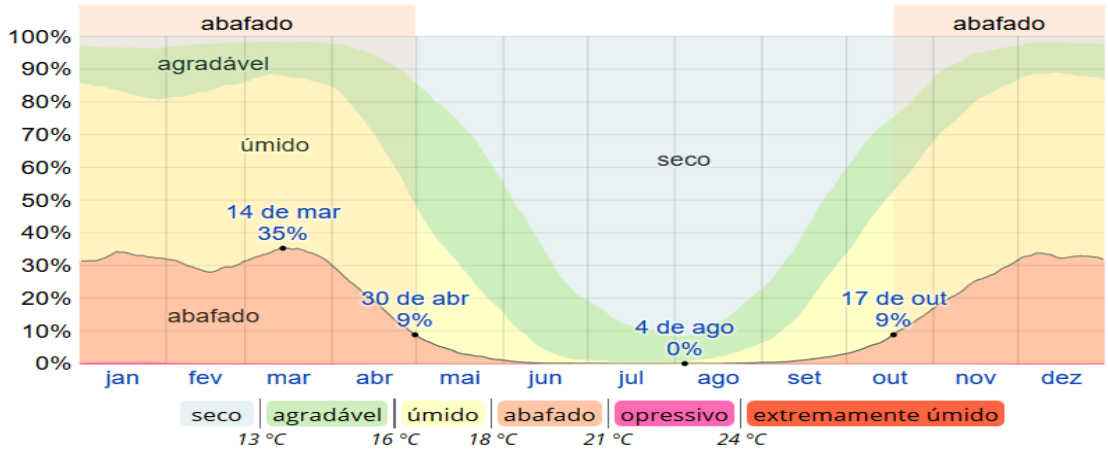
Neste capítulo, fizemos um estudo sobre a Umidade Relativa do Ar em Brasília, e propomos um modelo para tentar prever a possibilidade do próximo dia ter a umidade adequada, estar em estado de atenção, estado de alerta ou estado de emergência. Em um segundo momento, dividimos a série histórica dos dados de umidade relativa do ar em dois períodos e comparamos para poder entender os possíveis impactos da urbanização na umidade relativa do ar ao longo do tempo.

3.1 Umidade Relativa do Ar em Brasília

Brasília encontra-se no centro-oeste do território brasileiro, na latitude 16º sul e longitude 48º oeste a uma altitude média de 1100 metros em relação ao nível do mar. Quanto a classificação climática, Brasília possui um clima tropical de altitude que é caracterizado por grandes amplitudes diárias e duas estações bem definidas: quente e úmida no verão e seca no inverno.

A Figura abaixo apresenta a relação entre a umidade e o conforto respiratório para o Distrito Federal.

Figura 17 – Relação entre Umidade e Conforto Respiratório para o Distrito Federal



Fonte: www.weatherspark.com

3.2 Aplicação de Cadeias de Ordem Variável

Para a aplicação das Cadeias de Ordem Variável, utilizamos dados da umidade relativa do ar em Brasília. A umidade relativa do ar é a relação entre a quantidade de água presente no ar (medida em g/m^3) e a quantidade máxima que poderia haver na mesma temperatura. Os dados podem ser acessados através do portal INMET para a estação BRASILIA - DF (OMM:83377).

Os dados de umidade relativa do ar são medidos através de um instrumento chamado Psicrômetro, e são representados em percentual. Para a utilização das Cadeias de Ordem Variável, categorizamos os dados de umidade mínima diária por faixas conforme feito pelo Instituto Nacional de Meteorologia. O INMET define:

- Acima de 30%: Umidade mínima adequada para a saúde;
- Entre 20% e 30%: Estado de Atenção;
- Entre 12% e 20%: Estado de Alerta;
- Abaixo de 12%: Estado de Emergência.

Baseado nos resultados obtidos no capítulo anterior, escolhemos a constante de penalização $\delta = 0.13$ para a aplicação do estimador BIC nos dados de umidade mínima diária. Consideramos o alfabeto $\mathcal{A} = \{0, 1, 2, 3\}$ que representam respectivamente Estado de Emergência, Estado de Alerta, Estado de Atenção e Umidade Adequada.

A Figura 12 representa a árvore de Contexto estimada pelo estimador BIC com profundidade d = 3 considerando o período de 1962 à 2018.

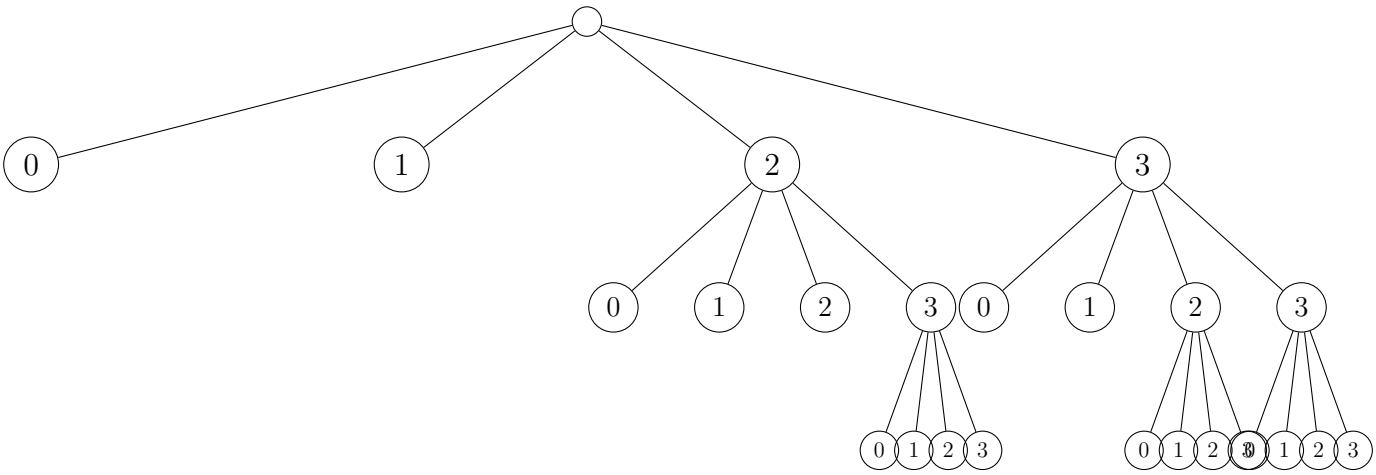


Figura 18 – Árvore de Contextos estimada, 1962 à 2018

Uma característica interessante dos modelos de ordem variável é a interpretação natural que a árvore de contextos gera. Observando a Figura 12, percebemos que para os dias em que a umidade é menor, podemos predizer o nível de umidade do próximo dia com base na informação de um dia atrás. Porém, para os dias mais úmidos, precisaríamos da informação do dia atual e dos dois dias anteriores.

Observando a Tabela 3, vemos que para uma sequência de dias mais úmidos a probabilidade do dia seguinte ser úmido é grande ao passo que a probabilidade de se ter vários dias úmidos seguidos de um dia seco é pequena. Para uma sequência de dias com muita umidade intercalada de sequidão, esperamos que o dia seguinte seja úmido também.

Tabela 5 – Probabilidades de Transição Estimadas

Contextos	**Probabilidades de Transição**			
	0	1	2	3
0	0,048	0,333	0,286	0,333
1	0,021	0,079	0,325	0,575
02	0,111	0,111	0,333	0,444
12	0,016	0,119	0,325	0,540
22	0,002	0,089	0,359	0,550
032	0,167	0,167	0,167	0,500
132	0,038	0,101	0,329	0,532
232	0,006	0,086	0,309	0,600
332	0,002	0,045	0,210	0,743
03	0,100	0,300	0,300	0,300
13	0,014	0,077	0,344	0,566
023	0,143	0,143	0,143	0,571
123	0,014	0,127	0,423	0,437
223	0,003	0,071	0,331	0,595
323	0,001	0,038	0,220	0,741
033	0,167	0,167	0,333	0,333
133	0,008	0,063	0,313	0,617
233	0,002	0,041	0,234	0,722
333	0,001	0,005	0,044	0,950

O crescimento urbano e consequentemente a alteração do solo (impermeabilização da superfície e verticalização) pode provocar uma alteração do clima. Considerando a evolução urbana de Brasília, fizemos um estudo comparativo entre dois períodos, na qual o ano climático de referência (ano de corte) foi o ano de 2000.

Uma característica dos modelos de Ordem Variável é que, a representação em forma de Árvores de Contexto nos permite comparar períodos diferentes com facilidade. Através disso conseguimos perceber se houve alguma alteração no processo ao longo do tempo. Para essa análise comparativa, separamos os dados de umidade relativa do ar em dois períodos: antes de 2000 e depois de 2000. Considerando o estimador BIC, $\delta = 0.13$ e d = 3, apresentamos as árvores obtidas a seguir.

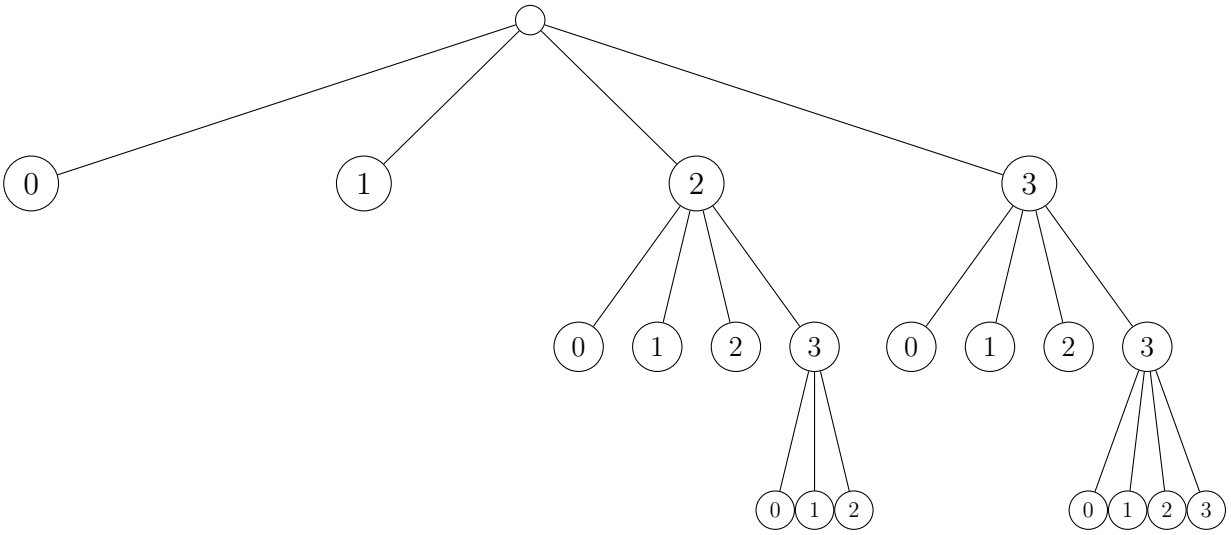


Figura 19 – Árvore de Contextos estimada, período anterior à 2000

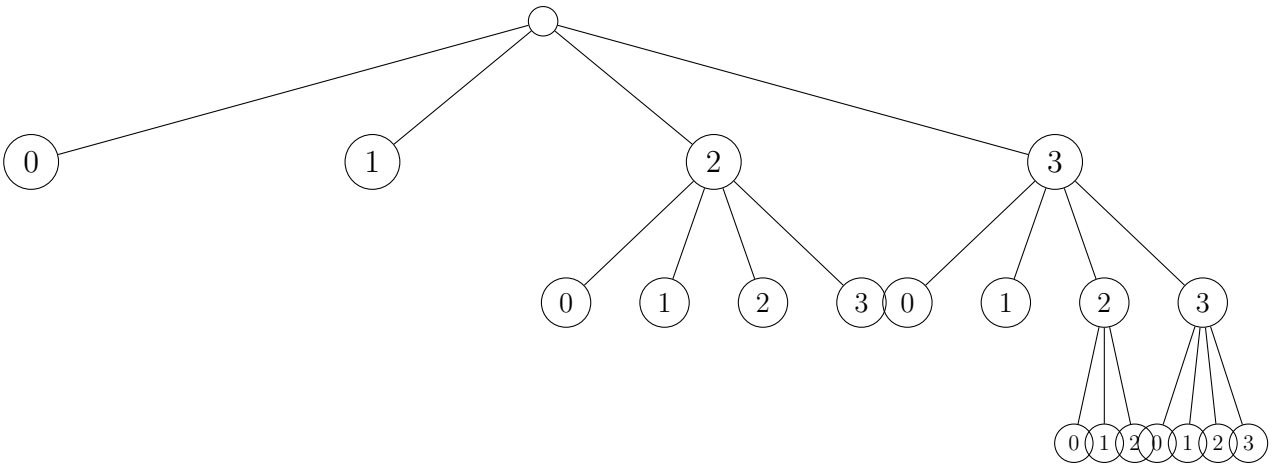


Figura 20 – Árvore de Contextos estimada, período posterior à 2000

Através das Figuras 19 e 20 podemos ver que o processo sofreu alteração ao longo do tempo. Na Figura 19, vemos que se um dia teve estado de atenção (2), precisamos olhar até dois dias para trás para poder predizer a umidade do próximo dia. Já para mesma situação após 2000, vemos que seria necessário olhar apenas um dia antes do dia atual.

Algumas diferenças no processo não aparacem na construção da árvore de contextos porém, podemos observá-las através das Probabilidades de Transição Estimadas. As Tabelas 6 e 7 trazem esses resultados.

Os contextos das árvores estimadas estão representados na primeira coluna de cada tabela enquanto o próximo elemento da sequência está no topo da tabela. A título de exemplo, peguemos a Tabela 6 que se refere aos dados de umidade relativa do ar para os anos anteriores a 2000. Vemos que, se tivermos três dias úmidos seguidos (representado pelo contexto 333) a probabilidade de ter um estado de emergência (0) no dia seguinte é quase 0. Por outro lado, a probabilidade de ter mais um dia úmido é $95,1\%$. Esses resultados são esperados visto que uma mudança brusca na umidade relativa do ar de um

dia para o outro não é natural até para um clima tropical como o de Brasília.

Os dias com estado de emergência são tão raros que a probabilidade de ter dois dias seguidos assim é abaixo de 6%. Para o período após 2000, vemos que essa probabilidade dobra, chegando a $12,5\%$.

De maneira geral as características de umidade de Brasília continuam as mesmas para o período antes de 2000 e após 2000, entretanto, verificamos algumas diferenças nas probabilidades de transição para os dias mais secos. Além disso, vimos que para os dias mais úmidos após 2000, precisamos de mais informação para predizer a umidade do próximo dia comparado com o período anterior a 2000. Esse aumento na incerteza pode ser causado por vários fatores como o aumento da emissão, ou acúmulo de monóxido de carbono em Brasília ao longo do tempo.

Tabela 6 – Probabilidades de Transição Estimadas, período anterior à 2000

Contextos	**Probabilidades de Transição**			
	0	1	2	3
0	0,059	0,412	0,353	0,176
1	0,018	0,091	0,312	0,580
2	0,111	0,111	0,333	0,444
12	0,022	0,146	0,337	0,494
22	0,002	0,098	0,378	0,522
132	0,050	0,100	0,367	0,483
232	0,009	0,091	0,307	0,593
332	0,002	0,050	0,208	0,740
3	0,167	0,333	0,167	0,333
13	0,018	0,104	0,350	0,528
23	0,001	0,046	0,249	0,704
033	0,200	0,200	0,200	0,400
133	0,011	0,056	0,337	0,596
233	0,003	0,042	0,230	0,725
333	0,001	0,003	0,045	0,951

Tabela 7 – Probabilidades de Transição Estimadas, período posterior à 2000

Contextos	**Probabilidades de Transição**			
	0	1	2	3
0	0,125	0,125	0,125	0,625
1	0,037	0,056	0,355	0,551
2	0,003	0,054	0,262	0,681
03	0,125	0,250	0,375	0,250
13	0,016	0,016	0,323	0,645
123	0,036	0,250	0,429	0,286
223	0,009	0,073	0,339	0,578
333	0,000	0,005	0,047	0,948
033	0,200	0,200	0,400	0,200
133	0,023	0,093	0,256	0,628
233	0,003	0,043	0,243	0,710
333	0,000	0,005	0,047	0,948

.

4 Considerações Finais

Neste trabalho, estudamos as Cadeias de Ordem Variável considerando um alfabeto $\mathcal{A} = \{0, 1, 2, 3\}$. Observamos a robustez dos estimadores BIC e Contexto Modificado quando simulamos amostras de tamanhos diferentes para processos distintos e mesmo assim tivemos um bom percentual de retornos corretos. Em um segundo momento, introduzimos contaminação nos dados simulados através do Modelo de Contaminação Zero Inflado e tentamos recuperar a árvore original do processo.

A análise de desempenho para o estimador tratado neste trabalho pode ser visto como um pequeno acréscimo a literatura sobre o tema. O estimador BIC mostrou-se muito robusto na análise em geral, principalmente para árvores incompletas e mais profundas. Para os processos com contaminação o estimador se mostrou sensível ao tamanho da amostra,desempenhando-se melhor em amostras maiores do que 5 mil.

Para a aplicação de Cadeias de Ordem Variável, modelamos dados de umidade relativa do ar em Brasília considerando o período de 1962 à 2018 e tentamos prever a possibilidade do próximo dia ter a umidade adequada para saúde, estar em estado de atenção, estado de alerta ou estado de emergência. Em um segundo momento, dividimos a série histórica em dois períodos e tentamos verificar se houve alguma mudança na umidade relativa do ar em Brasília devido ao processo de urbanização ao longo do tempo.

Verificamos que de maneira geral as características de umidade do Brasília continuam as mesmas para o período antes de 2000 e após 2000, entretanto, verificamos algumas diferenças nas probabilidades de transição para os dias mais secos. Observamos que para os dados anteriores a 2000, precisávamos de menos informação para predizer a umidade do próximo dia, ao passo que para o período após 2000, temos uma incerteza maior na predição. Esse aumento na incerteza pode ser causado por vários fatores como o aumento da emissão, ou acúmulo de monóxido de carbono em Brasília ao longo do tempo.

5 Referências Bibliográficas

[1] Csiszár, I. and Talata, Z., Context tree estimation for not necessarily nite memory processes, via BIC and MDL, IEEE Trans. Inform. Theory 52, Number3, 1007-1016, 2006

[2] Galves, Antonio. Leonardi, Florencia., Exponential inequalities for empirical unbounded context trees. Progress in Probability 60 (2008), 257-270.

[3] Garcia, N. L., Moreira, L., Stochastically Perturbed Chains of Variable Memory, Springer, New York, 2015

[4] Rissanen, J., A universal data compression system, IEEE Trans. Inform. Theory 29(5): 656-664, (1983).

[5] Matta, David. Algoritmos de estimação para Cadeias de Markov de Alcance Variável - aplicações a detecção do ritmo em textos escritos. 2008. 87p.Dissertação de Mestrado (Mestre em Estatística) - Instituto de Matemática,Estatística e Computação Cientifica, Universidade Estadual de Campinas, Campinas.

[6] Moreira, Lucas. Processos de Ordem Infinita Estocasticamente Perturbados. 2012. 54 p. Tese (Doutorado em Estatística) - Instituto de Matemática, Estatística e Computação Cientifica, Universidade Estadual de Campinas, Campinas.

[7] Moreira, L.; Quintino, F. S. . CADEIAS DE ORDEM VARIÁVEL ESTOCASTICAMENTE PERTURBADAS: APLICAÇÃO EM MODELOS CLIMÁTICOS. Revista SODEBRAS, v. 10, p. 141-146, 2015.

[8] Maciel, Alexandra. Projeto Bioclimático em Brasília: Estudo de Caso em Edifícios de Escritórios - Curso de pós-graduação em engenharia civil, Universidade Federal de Santa Catarina.

6 Apêndice

```
 ################################## FUNÇÃO BIC ####################################
fbic <- function(dados,d,constante,perturbacao = 0){

 # perturbação

  if (perturbacao != 0){
    for (i in 1:length(dados)){
      dados[i] <- dados[i]*rbinom(1,1,1-perturbacao)
    }
  }

  # valor de |A| (considerando A = {0,1,...,|A| - 1})

  alfabeto <- max(dados)+1

  # função para converter (i,j) em sequência

  fconverte <- function(i,j){
    conversao <- character(1)
    for (l in (d+1-j):1){
      conversao <- paste(floor((i-1)/alfabeto^(l-1)),conversao,sep="")
      i <- ((i-1) %% alfabeto^(l-1)) + 1
    }
    conversao
  }

  # função para completar matriz

  fcompleta <- function(matriz,q) {
    d <- (ncol(matriz)-1)
    for (j in 1:d){
      for (i in 1:(q^d)){
        matriz[floor((i-1)/q)+1,j+1] <- matriz[floor((i-1)/q)+1,j+1]
        + matriz[i,j]
      }
```

```
  }
  matriz
}

# contagem de N_n(s,a) e N_n(s)

num <- array(0,c(alfabeto^d,d+1,alfabeto))

for (tempo in (d+1):length(dados)){
  i <- 0
  ajuste <- 0
  for (passado in (tempo-d):(tempo-1)){
    i <- i + (alfabeto^ajuste)*(dados[passado])
    ajuste <- ajuste + 1
  }
  num[i+1,1,(dados[tempo])+1] <- num[i+1,1,(dados[tempo])+1] + 1
}

for (i in 1:alfabeto){
  num[,,i] <- fcompleta(num[,,i],alfabeto)
}

numt <- apply(num,c(1,2),sum)

# probabilidades de transição estimadas

tr <- array(0,c(alfabeto^d,d+1,alfabeto))

for (j in 1:(d+1)){
  for (i in 1:alfabeto^(d+1-j)){
    for (k in 1:alfabeto){
      if (numt[i,j] == 0) tr[i,j,k] <- 1/alfabeto
      else tr[i,j,k] <- num[i,j,k]/numt[i,j]
    }
  }
}

# achando a matriz v e a matriz x
```

```
matriz <- matrix(0,alfabeto^d,d+1)
matrizv <- matrix(0,alfabeto^d,d+1)
matrizx <- matrix(0,alfabeto^d,d+1)

for (j in 1:(d+1)){
  for (i in 1:alfabeto^(d+1-j)){
    for (k in 1:alfabeto){
      if (tr[i,j,k] != 0){
        matriz[i,j] <- matriz[i,j] + num[i,j,k]*log(tr[i,j,k])
      }
      matriz[i,j] <- matriz[i,j] - constante*log(length(dados))
      if (numt[i,j] == 0) matriz[i,j] <- 0
    }
  }
}

for (i in 1:alfabeto^d){
  matrizv[i,1] <- matriz[i,1]
}
for (j in 2:(d+1)){
  for (i in 1:alfabeto^(d+1-j)){
    for (anterior in 1:alfabeto){
      matrizv[i,j] <- matrizv[i,j] + matriz[(i-1)*alfabeto+anterior,j-1]
    }
    if (matriz[i,j] > matrizv[i,j]){
      matrizv[i,j] <- matriz[i,j]
    }
  }
}

for (j in 1:(d+1)){
  for (i in 1:alfabeto^(d+1-j)){
    matrizx[i,j] <- as.integer(matrizv[i,j] > matriz[i,j])
  }
}

# achando a árvore

arvore <- character()
```

```
  arvore[1] <- "sequencia vazia"
  index <- 1
  valor <- 0
  for (i in 1:alfabeto^d){
    for (j in 1:d){
      valor <- 0
      sufixo <- i
      while (matrizx[i,j] == 0 &&
      matrizx[floor((sufixo-1)/alfabeto)+1,j+valor+1] == 1){
        valor <- valor + 1
        sufixo <- floor((sufixo-1)/alfabeto+1)
        if (j+valor > d) break
      }
      if (valor == d-j+1 && numt[i,j] > 0){
        arvore[index] <- fconverte(i,j)
        index <- index + 1
      }
    }
  }
 arvore
}

################### Exemplo de Simulação com uma Árvore Construída ###################

a0=0.25
b0=0.45
c0=0.65

a1=0.17
b1=0.38
c1=0.75

a32=0.28
b32=0.72
c32=0.9

a22=0.28
b22=0.72
c22=0.9
```

```
a12=0.56
b12=0.60
c12=0.7

a02=0.18
b02=0.34
c02=0.65

a03=0.35
b03=.65
c03=0.78

a13=0.35
b13=0.44
c13=0.96


a23=0.23
b23=.65
c23=0.89

a33=0.52
b33=.72
c33=0.96

a033=0.68
b033=.65
c033=0.82

a133=0.54
b133=.52
c133=0.69

a233=0.52
b233=.72
c233=0.96

a333=0.52
```

```
b333=.72
c333=0.96

tamanho = 1000
dados = vector("integer", length = tamanho)
dados[c(1,2,3,4)] = c(0,1,2,3)
dados = rep(dados, times = 100)
dados2 = split(dados, ceiling(seq_along(dados)/tamanho))
sequ = seq(0, 1, 0.01)
lista.bic = rep(list(list()), length(sequ))
n.amostras=100

for (i in 5:tamanho) {
  for(k in 1:n.amostras) {
    v = runif(1)
    if(dados2[[k]][[i-1]] == 0){
      if(v < a0) {
        dados2[[k]][[i]] = 0
      } else if(v < b0) {
        dados2[[k]][[i]] = 1
      } else if(v < c0) {
        dados2[[k]][[i]] = 2
      } else {
        dados2[[k]][[i]] = 3
      }
    }
    if(dados2[[k]][[i-1]] == 1){
      if(v < a1) {
        dados2[[k]][[i]] = 0
      } else if(v < b1) {
        dados2[[k]][[i]] = 1
      } else if(v < c1) {
        dados2[[k]][[i]] = 2
      } else {
        dados2[[k]][[i]] = 3
      }
    }

    if(dados2[[k]][[i-2]] == 0 && dados2[[k]][[i-1]] == 2){
```

```
  if(v < a02) {
    dados2[[k]][[i]] = 0
  } else if(v < b02) {
    dados2[[k]][[i]] = 1
  } else if(v < c02) {
    dados2[[k]][[i]] = 2
  } else {
    dados2[[k]][[i]] = 3
  }
}
if(dados2[[k]][[i-2]] == 1 && dados2[[k]][[i-1]] == 2){
  if(v < a12) {
    dados2[[k]][[i]] = 0
  } else if(v < b12) {
    dados2[[k]][[i]] = 1
  } else if(v < c12) {
    dados2[[k]][[i]] = 2
  } else {
    dados2[[k]][[i]] = 3
  }
}
if(dados2[[k]][[i-2]] == 2 && dados2[[k]][[i-1]] == 2){
  if(v < a22) {
    dados2[[k]][[i]] = 0
  } else if(v < b22) {
    dados2[[k]][[i]] = 1
  } else if(v < c22) {
    dados2[[k]][[i]] = 2
  } else {
    dados2[[k]][[i]] = 3
  }
}

if(dados2[[k]][[i-2]] == 3 && dados2[[k]][[i-1]] == 2){
  if(v < a32) {
    dados2[[k]][[i]] = 0
  } else if(v < b32) {
    dados2[[k]][[i]] = 1
  } else if(v < c32) {
```

```
      dados2[[k]][[i]] = 2
    } else {
      dados2[[k]][[i]] = 3
    }
  }
  if(dados2[[k]][[i-2]] == 0 && dados2[[k]][[i-1]] == 3){
    if(v < a03) {
      dados2[[k]][[i]] = 0
    } else if(v < b03) {
      dados2[[k]][[i]] = 1
    } else if(v < c03) {
      dados2[[k]][[i]] = 2
    } else {
      dados2[[k]][[i]] = 3
    }
  }

  if(dados2[[k]][[i-2]] == 1 && dados2[[k]][[i-1]] == 3){
    if(v < a13) {
      dados2[[k]][[i]] = 0
    } else if(v < b13) {
      dados2[[k]][[i]] = 1
    } else if(v < c13) {
      dados2[[k]][[i]] = 2
    } else {
      dados2[[k]][[i]] = 3
    }
  }

  if(dados2[[k]][[i-2]] == 2 && dados2[[k]][[i-1]] == 3){
    if(v < a23) {
      dados2[[k]][[i]] = 0
    } else if(v < b23) {
      dados2[[k]][[i]] = 1
    } else if(v < c23) {
      dados2[[k]][[i]] = 2
    } else {
      dados2[[k]][[i]] = 3
    }
```

```
}

if(dados2[[k]][[i-2]] == 3 && dados2[[k]][[i-1]] == 3){
  if(v < a33) {
    dados2[[k]][[i]] = 0
  } else if(v < b33) {
    dados2[[k]][[i]] = 1
  } else if(v < c33) {
    dados2[[k]][[i]] = 2
  } else {
    dados2[[k]][[i]] = 3
  }
}
if(dados2[[k]][[i-3]] == 0 && dados2[[k]][[i-2]] == 3 && dados2[[k]][[i-1]] ==
  if(v < a033) {
    dados2[[k]][[i]] = 0
  } else if(v < b033) {
    dados2[[k]][[i]] = 1
  } else if(v < c033) {
    dados2[[k]][[i]] = 2
  } else {
    dados2[[k]][[i]] = 3
  }
}
if(dados2[[k]][[i-3]] == 1 && dados2[[k]][[i-2]] == 3 && dados2[[k]][[i-1]] ==
  if(v < a133) {
    dados2[[k]][[i]] = 0
  } else if(v < b133) {
    dados2[[k]][[i]] = 1
  } else if(v < c133) {
    dados2[[k]][[i]] = 2
  }  else {
    dados2[[k]][[i]] = 3
  }
}
if(dados2[[k]][[i-3]] == 2 && dados2[[k]][[i-2]] == 3 && dados2[[k]][[i-1]] ==
  if(v < a233) {
    dados2[[k]][[i]] = 0
  } else if(v < b233) {
```

```
        dados2[[k]][[i]] = 1
      } else if(v < c233) {
        dados2[[k]][[i]] = 2
      } else {
        dados2[[k]][[i]] = 3
      }
    }
    if(dados2[[k]][[i-3]] == 3 && dados2[[k]][[i-2]] == 3 && dados2[[k]][[i-1]] == 3){
      if(v < a333) {
        dados2[[k]][[i]] = 0
      } else if(v < b333) {
        dados2[[k]][[i]] = 1
      } else if(v < c333) {
        dados2[[k]][[i]] = 2
      } else {
        dados2[[k]][[i]] = 3
      }
    }

  }

}


P = c("0","1","02","12","22","32","03","13","23","033","133","233","333")  #Contextos

for (j in 1:100) {
  for (i in 1:length(sequ)) {
    lista.bic[[i]][[j]] = fbic(dados2[[j]],d = 3,constante = sequ[i])
  }
}

comparacoes <- rep(list(list()), length(sequ))


for(i in 1:length(sequ)){
  for(j in 1:100)
    comparacoes[[i]][[j]] = identical(lista.bic[[i]][[j]], P)
}
```

```
acertos = rep(list(list()), length(sequ))


for(i in 1:length(sequ)){
  acertos[[i]] = unlist(comparacoes[[i]])
}

for(i in 1:length(sequ)){
  acertos[[i]] = sum(acertos[[i]], na.rm = T)
}


bic_1k = unlist(acertos)
names(bic_1k) = sequ
bic_1k=cbind(bic_1k)/100
bic_limpo_1k               # Proporção de acertos para cada
                            #constante de poda simulada.
```