Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Estatística

Aplicações de Cadeias de Ordem Variável Estocasticamente Perturbadas

Felipe Sousa Quintino

Trabalho de Conclusão de Curso apresentado ao Departamento de Estatística da Universidade de Brasília, como parte dos requisitos para a obtenção do título de Bacharel em Estatística.

Brasília
2015

FELIPE SOUSA QUINTINO

Aplicações de Cadeias de Ordem Variável Estocasticamente Perturbadas

Trabalho de Conclusão de Curso apresentado ao Departamento de Estatística da Universidade de Brasília, como parte dos requisitos para a obtenção do título de Bacharel em Estatística.

Orientador: Prof. Dr. **Lucas Moreira**

Brasília
2015

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Estatística

Aplicações de Cadeias de Ordem Variável Estocasticamente Perturbadas

por

Felipe Sousa Quintino

Monografia apresentada ao Departamento de Estatística da Universidade de Brasília, como parte dos requisitos para obtenção do grau de

BACHAREL EM ESTATÍSTICA

Brasília, 26 de 06 de 2015.

Banca Examinadora:

Prof. Dr. Lucas Moreira - Orientador (EST/UnB)

Profa. Dra. Cátia Regina Gonçalves - Membro (MAT/UnB)

Prof. PhD Gustavo L. Gilardone Avalle - Membro (EST/UnB)

Dedicatória

Dedico este trabalho à meus pais,
por todo amor que sempre me dedicaram,
por todos ensinamentos e apoio,
à vocês que são meus maiores exemplos...

Agradecimentos

Agradeço primeiramente à Deus por todas as oportunidades que tem me proporcionado até aqui. Agradeço também a minha família que sempre esteve ao meu lado nos momentos em que precisei. De forma especial agradeço a Elci (minha mãe) e Aline (minha namorada), pois, foram aquelas que mais me cobraram e apoiaram para que eu superasse cada obstáculo.

Agradeço ao meu orientador, o professor Dr. Lucas Moreira, por toda a disposição para prontamente me ajudar com qualquer dificuldade que tive ao longo dos estudos e da elaboração desse trabalho e por todos os seus ensinamentos, motivações e cobranças.

Agradeço a todos os amigos que fiz ao longo do curso, tanto aos da Estatística como aos de outros cursos, pois, em muito contribuíram para a minha formação. Não citarei todos para não ser injusto esquecendo vários nomes, mas em especial à Agda, Alex, Augusto, Erique, Felipe, Geisiane, Guilherme e Thiago.

Agradeço aos professores dos departamentos de Estatística e Matemática dos quais tive o prazer de ser aluno. Em especial as professoras Claudete e Maria Amélia que muito me ensinaram e aconselharam ao longo de minha graduação. Também agradeço a professora Dra. Cátia Regina Gonçalves e ao professor PhD Gustavo Leonel Gilardone Avalle por aceitarem participar da Banca Examinadora desse trabalho.

Resumo

Neste trabalho estudamos a estimação das árvores de contextos de Cadeias de Ordem Variável utilizando uma amostra perturbada do processo segundo algum dos três modelos de contaminação estudados neste trabalho. No primeiro modelo consideramos uma Cadeia de Ordem Variável com alfabeto binário em que, a cada instante de tempo, um dos símbolos pode ser modificado com uma probabilidade pequena e fixada. No segundo modelo consideramos uma cadeia com alfabeto binário em que, a cada instante de tempo, o processo perturbado assume aleatoriamente o valor da cadeia original ou uma função que depende deste valor, com probabilidade pequena e fixada. No terceiro modelo consideramos duas Cadeias de Ordem Variável independentes, tomando valores num mesmo alfabeto finito, onde o processo perturbado assume aleatoriamente, a cada instante de tempo, um dos dois processos originais com uma probabilidade grande e fixa. Os modelos de contaminação foram comparados através das simulações de amostras perturbados de processos. Pela simplicidade do primeiro modelo de contaminação foi possível recuperar a árvore de contextos do processo original mesmo com alta probabilidade de contaminação. Enquanto utilizando os outros dois modelos recuperamos a árvore de contextos do processo original apenas quando a probabilidade de perturbação era suficientemente pequena. Em seguida, propomos modelos meteorológicos para prever a possibilidade de o próximo dia ser quente ou não, dado as informações de temperaturas máximas dos dias anteriores.

Palavras-chave: Cadeias de Ordem Variável, Árvores de Contextos, Modelos de Contaminação estocástica, Modelos Meteorológicos.

Abstract

We analyzed the estimation of the context trees of the Chains of Variable Memory using a perturbed sample of the process according to one of the three contamination models studied in this paper. In the first model, we examined a Chain of Variable Memory with binary alphabet in which, at every moment, one of the symbols might be modified with a small and fixed probability. In the second model, we examined a chain with binary alphabet in which, at every moment, the disturbed process randomly assumes the value of the original chain or a function that depends on this value, with a small and fixed probability. In the third model, we examined two independent Chains of Variable Memory taking values in the same finite alphabet, in which the disturbed process randomly assumes, at every moment, one of the two original processes with a large and fixed probability. The contamination models were compared by means of the simulations of the processes perturbed samples. Due to the simplicity of the first contamination model, it was possible to recover the context tree of the original process in spite of the high contamination probability; whereas, when we used the two other models, we recovered the context tree of the original process only when the disturbance probability was sufficiently small. We then proposed meteorological models to predict the possibility of the following day being hot, given the information of maximum temperatures of the previous days.

Key Words: Chains of Variable Memory, Context Trees, Contamination Stochastic Models, Meteorological Models.

Sumário

Lista de Figuras

Lista de Tabelas

Introdução

A motivação original deste trabalho foi estudar modelos de contaminação estocástica. Especificamente, considerando os modelos de contaminação, apresentados em Collet, Galves e Leonardi (2008) e por Garcia e Moreira (2015), propomos modelos meteorológicos para predizer a possibilidade de o próximo dia ser quente ou não dado às informações de temperaturas máximas dos dias anteriores. O desenvolvimento do trabalho foi realizado dentro do cenário da Teoria de Processos Estocásticos e com enfoque de Probabilidade Clássica. Toda a inferência foi baseada numa versão do Algoritmo Contexto introduzido em Galves, Maume-Deschamps e Schmitt (2006) para árvores finitas e estendido para árvores ilimitadas em Galves e Leonardi (2008).

Os modelos de ordem variável foram introduzidas em Rissanen (1983) e chamados fontes de memória finita ou máquinas de árvores, em que a porção do passado necessário para predizer o próximo símbolo não é fixa, mas é uma função da sequência dos símbolos passados. Na literatura estatística recente, estes modelos são chamados *Cadeias de Ordem Variável*.

Rissanen (1983) chamou de *contexto* a porção do passado necessária para predizer o próximo simbolo. O conjunto de todos os contextos pode ser representado por uma árvore probabilística com raiz e rótulos chamada de *árvore de contextos* do processo. Em seu trabalho, Rissanem estudou a cadeias de ordem finita. No entanto, a extensão de um modelo com ordem variável para uma situação não Markoviana, em que os contextos são ainda finitos, porém com comprimento ilimitado, ocorre naturalmente. Com a leitura de Galves e Löcherbach (2008) é possível fazer um levantamento recente acerca do tema.

Um aspecto vantajoso dos modelos de ordem variável, em relação as Cadeias de Markov de ordem fixa, é a redução do número de parâmetros a serem estimados. Isto ocorre, pois, os modelos de ordem variável levam em conta as dependências estruturais presentes nos dados. Outra característica interessante é que em muitas aplicações, a forma da árvore de contextos tem uma interpretação natural e informativa.

Além de introduzir as Cadeias de Ordem Variável, Rissanen (1983) também propôs um algoritmo para estimar a árvore de contextos, chamado *Algoritmo Contexto*.

Diversos estudos recentes abordaram a questão da estimação da árvore de contextos para Cadeias de Ordem Variável bem como o correspondente conjunto associado de probabilidades de transição, utilizando variantes do Algoritmo Contexto de Rissanen (1983). Dentre eles destacam-se Bühlmann e Wyner (1999) para o caso onde a ordem da cadeia é limitada, Ferrari e Wyner (2003) para ordem não limitada, o BIC de Csiszar e Talata (2005) e também Duarte et al. (2006) que deram uma majoração para a velocidade de convergência do Algoritmo Contexto para Cadeias de Ordem Variável não limitadas.

Collet, Galves e Leonardi (2008) propuseram um Modelo de Contaminação Estocástica considerando uma Cadeia de Ordem Variável com alfabeto binário em que, a cada instante de tempo, o processo perturbado assume aleatoriamente o valor da cadeia original ou uma função que depende deste valor, com probabilidade pequena e fixada. Além do modelo proposto, provaram que é possível recuperar a árvore de contextos do processo original através de uma amostra contaminada segundo este modelo.

Garcia e Moreira (2015) apresentaram dois Modelos de Contaminação Estocástica. No primeiro modelo consideraram uma Cadeia de Ordem Variável com alfabeto binário em que, a cada instante de tempo, um dos símbolos pode ser modificado com uma probabilidade pequena e fixada. No segundo modelo consideraram duas Cadeias de Ordem Variável independentes, tomando valores num mesmo alfabeto finito, o processo perturbado assume aleatoriamente, a cada instante de tempo, um dos dois processos originais com uma probabilidade grande e fixa. Garcia e Moreira (2015) também provaram que utilizando uma amostra contaminada segundo estes modelos de contaminação é possível recuperar a árvore de contextos do processo original.

Dessa forma, através do estudo de simulações verificamos o bom desempenho da versão do Algoritmo Contexto proposta em Galves e Leonardi (2008) quando utilizamos amostras contaminadas. Todas as simulações e estimativas foram realizadas através do ambiente R de computação estatística (R Core Team, 2014).

O presente estudo está organizado da seguinte forma: no Capítulo 1 apresentamos as notações e definições básicas. No Capítulo 2 definimos a versão do Algoritmo Contexto utilizada para estimação da árvore de contextos, assim como os modelos de contaminação considerados nesse trabalho. No Capítulo 3 apresentamos e discutidos os resultados obtidos através das simulações de processos contaminados e da aplicação dos modelos de contaminação em dados meteorológicos, sendo comparados com uma modelagem por Cadeia de Markov de ordem k. O Capítulo 4 traz as conclusões do trabalho. Nos Apêndices A e B apresentamos, respectivamente, os códigos desenvolvidos em ambiente R do estimador de árvore de contextos e do Algoritmo de Contaminação da amostra.

Este trabalho foi parte de um projeto do Programa de Iniciação Científica da Universidade de Brasília (ProIC/DPP/UnB) CNPq 2014/2015.

Capítulo 1

Revisão Bibliográfica

Neste capítulo definimos formalmente uma Cadeia de Ordem Variável e, para conveniência do leitor, fizemos uma breve revisão das notações e definições que assumimos para o processo.

1.1 Notações e Definições

Considere o alfabeto $\mathcal{A} = \{0, 1, \cdots, N-1\}$ com tamanho $|\mathcal{A}| = N$. Dados dois inteiros $m \leq n$ denotamos a_m^n a sequência de símbolos $a_m, a_{m+1}, \cdots, a_n$ de $\mathcal{A}$ e $\mathcal{A}_m^n$ o conjunto de tais sequências. O comprimento da sequência será $l(a_m^n) = n - m + 1$. Caso $n < m$, $a_m^n = \varnothing$ e $l(a_m^n) = 0$.

O conjunto de todas as sequências semi-infinitas e o conjunto de todas as sequências de símbolos de tamanho finito são denotados, respectivamente, por

$$\mathcal{A}_{-\infty}^{-1} = \mathcal{A}^{\{\cdots,-2,-1\}} \quad e \quad \mathcal{A}^* = \bigcup_{j=0}^{\infty} \mathcal{A}_{-j}^{-1},$$

em que para $j = 0$ corresponde ao conjunto das sequências vazias $\varnothing$.

Dadas duas sequências ω e v, com $l(\omega) < +\infty$, denotamos por $v\omega$ a sequência de comprimento $l(v) + l(\omega)$ obtida pela concatenação das duas sequências. Por exemplo, para $v = \cdots, v_{-n-2}, v_{-n-1}$ e $\omega = \omega_{-n}, \cdots, \omega_{-2}, \omega_{-1}$, a sequência obtida pela concatenação de v e ω será $v\omega = \cdots, v_{-n-2}, v_{-n-1}, \omega_{-n}, \cdots, \omega_{-2}, \omega_{-1}$. Note que, para o caso em que $v = \varnothing$ obtêm-se $v\omega = \varnothing\omega = \omega$. Analogamente ocorre para $\omega = \varnothing$.

Uma sequência u é dita ser um *sufixo* de ω se existir s, com $l(s) \geq 1$, tal que $\omega = su$ e será denotada por $u \prec \omega$. Caso $u \prec \omega$ ou $u = \omega$, será denotado por $u \preceq \omega$. Dada uma sequência finita ω denotamos por suf(ω) o maior sufixo de ω.

Ao longo desse trabalho consideramos o processo $\mathbf{X} = \{X_t,\ t \in \mathbb{Z}\}$ estacionário e ergódico sobre o alfabeto $\mathcal{A} = \{0, 1, \cdots, N-1\}$. Assumimos que o processo

$\mathbf{X}$ é compatível com a probabilidade de transição $p_X(.|.)$, ou seja,

$$p_X(a|\omega) = \mathbb{P}(X_0 = a|X_{-1} = \omega_{-1}, X_{-2} = \omega_{-2}, \cdots), \tag{1.1}$$

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 será denotada por

$$\mu_X(\omega) = \mathbb{P}(X_{-j}^{-1} = \omega). \tag{1.2}$$

Com intuito de estimarmos a árvore de contextos de um processo $\mathbf{X}$, dada uma amostra contaminada desse processo, consideramos que $\mathbf{X}$ satisfaz as seguintes definições.

Definição 1.1 *Dizemos que um processo* $\mathbf{X}$ *é não-nulo se satisfaz*

$$\alpha_X = \inf\{p_X(a|\omega) : \ a \in \mathcal{A}, \ \omega \in \mathcal{A}_{-\infty}^{-1}\} > 0. \tag{1.3}$$

Definição 1.2 *Dizemos que um processo* $\mathbf{X}$ *possui taxa de continuidade somável se*

$$\beta_X = \sum_{k \in \mathbb{N}} \beta_{k,X} < +\infty, \tag{1.4}$$

em que a sequência $\{\beta_{k,X}\}_{k\in\mathbb{N}}$ *é definida por*

$$\beta_{k,X} := sup\left\{\left|1 - \frac{p_X(a|\omega)}{p_X(a|v)}\right| : \ a \in \mathcal{A}, \ v, \omega \in \mathcal{A}_{-\infty}^{-1} \ \ com \ \ \omega_{-k}^{-1} = v_{-k}^{-1}\right\}. \tag{1.5}$$

A sequência $\{\beta_{k,X}\}_{k\in\mathbb{N}}$ é chamada *taxa de continuidade do processo* $\mathbf{X}$. Note que, a condição de não-nulidade do processo $\mathbf{X}$ é necessária para que possamos definir a taxa de continuidade do processo por (1.5). A taxa de continuidade é uma propriedade esperada para o processo $\mathbf{X}$, pois, desejamos que dois passados coincidindo nos últimos k símbolos tenham a mesma influência na predição do próximo símbolo da sequência, a medida que k cresce.

Rissanen (1983) chamou de *contexto* a porção do passado necessária para predizer o próximo símbolo do processo, sendo o tamanho desta sequência é função do próprio passado. Um contexto infinito é uma sequência semi infinita tal que nenhum dos seus sufixos é um contexto. O conjunto de todos os contextos satisfaz a propriedade do sufixo, isto é, nenhum contexto é sufixo de outro contexto. Esta propriedade permite representar o conjunto de todos os contextos (finito ou infinito enumerável) como uma árvore probabilística com raiz e rótulos. Esta árvore é chamada *árvore de contextos* do processo $\mathbf{X}$. A seguir definiremos de maneira mais formal um contexto.

Definição 1.3 *Dizemos que uma sequência $\omega \in \mathcal{A}_{-j}^{-1}$ é um contexto do processo $\mathbf{X}$ se para toda sequência semi-infinita $x_{-\infty}^{-1} \in \mathcal{A}_{-\infty}^{-1}$ tendo ω como sufixo satisfazer*

$$\mathbb{P}(X_0 = a | X_{-\infty}^{-1} = x_{-\infty}^{-1}) = p_X(a|\omega), \quad \forall a \in \mathcal{A}, \tag{1.6}$$

e nenhum sufixo de ω satisfaz esta equação.

Denotamos por $d(\mathcal{T})$ a profundidade da árvore $\mathcal{T}$, ou seja,

$$d(\mathcal{T}) := \max\{l(\omega) : \omega \in \mathcal{T}\}.$$

Uma árvore $\mathcal{T}$ é dita *completa* se qualquer sequência em $\mathcal{A}_{-\infty}^{-1}$ pertence a $\mathcal{T}$ ou tem sufixo que pertence a $\mathcal{T}$. Dizemos que a árvore de contextos é *limitada* se o comprimento do maior contexto é finito. Caso contrário, $\mathcal{T}$ é dita *ilimitada*.

Dizemos que uma árvore é *irredutível* se nenhuma sequência pode ser substituída por um sufixo sem violar a propriedade sufixo. Essa noção foi introduzida em Csiszár e Talata (2006) e generaliza o conceito de árvore completa.

A seguir definiremos de maneira mais formal uma *árvore probabilística de contextos* e uma Cadeia de Ordem Variável.

Definição 1.4 *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) $\bar{p} = \{\bar{p}(.|\omega), \omega \in \mathcal{T}\}$ é uma família de probabilidades de transição sobre $\mathcal{A}$.

Definição 1.5 *Dizemos que o processo $\mathbf{X}$ é compatível com a árvore probabilística de contextos $(\mathcal{T}, \bar{p})$ se satisfaz*

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

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

Se $\mathbf{X}$ é compatível com a árvore probabilística de contextos $(\mathcal{T}, \bar{p})$, dizemos que $\mathbf{X}$ é uma *Cadeia de Ordem Variável* e denotamos a árvore de contextos de $\mathbf{X}$ por $\mathcal{T}_{\mathbf{X}}$. Note que em (1.4) da Definição (1.2), se $d(\mathcal{T}_{\mathbf{X}}) < +\infty$, então $\beta_{k,X} = 0$ para $k \geq d(\mathcal{T}_{\mathbf{X}})$, ou seja,

$$\beta_X = \sum_{k=0}^{d(\mathcal{T}_{\mathbf{X}})-1} \beta_{k,X} < +\infty.$$

Em alguns casos podemos estar interessados não na árvore de contextos do processo $\mathbf{X}$ mas na utilização desta árvore com uma restrição no tamanho da maior sequência. Seja K esta restrição. Neste caso, chamaremos de árvore truncada no nível K. Dessa forma, se definirmos $K \geq d(\mathcal{T}_{\mathbf{X}})$, estaremos considerando a própria árvore de contextos do processo $\mathbf{X}$.

Definição 1.6 *Dado um inteiro K, defina a árvore de contextos truncada no nível K por*

$$\mathcal{T}_{\mathbf{X}}|_K = \{\omega \in \mathcal{T}_{\mathbf{X}} : l(\omega) \leq K\} \cup \{\omega : l(\omega) = K \quad e \quad \omega \prec u, \ \textit{para algum} \ \ u \in \mathcal{T}_{\mathbf{X}}\}.$$

Considere $\mathbf{Z} = \{Z_t, \ t \in \mathbb{Z}\}$ um processo tomando valores num alfabeto finito $\mathcal{A} = \{0, 1, \cdots, N-1\}$. Seja $Z_1, \cdots, Z_n$ uma amostra aleatória do processo $\mathbf{Z}$. Para toda sequência finita ω, com $l(\omega) \leq n$, denotamos por $N_n(\omega)$ o número de vezes que observou-se a sequência ω na amostra, ou seja,

$$N_n(\omega) = \sum_{t=0}^{n-l(w)} \mathbf{1}_{\{Z_{t+1}^{t+l(\omega)} = \omega\}}. \tag{1.7}$$

Para todo elemento $a \in \mathcal{A}$ e para toda sequência finita ω, a probabilidade de transição empírica é dada por

$$\hat{p}_Z(a|\omega)_n = \frac{N_n(\omega a) + 1}{N_n(\omega .) + |\mathcal{A}|}. \tag{1.8}$$

Observe que a definição de $\hat{p}_Z(a|\omega)_n$ é conveniente, pois, é assintoticamente equivalente ao Estimador de Máxima Verossimilhança que é $\frac{N(\omega a)}{N(\omega .)}$ e evita-se uma definição adicional no caso $N(\omega \cdot) = 0$.

Antes de apresentar o estimador da árvore de contextos, definido no Capítulo 2, é necessário definirmos o seguinte operador

$$\Delta_n(\omega) := \max_{a \in \mathcal{A}} |\hat{p}_Z(a|\omega)_n - \hat{p}_Z(a|suf(\omega))_n|, \tag{1.9}$$

para qualquer sequência finita $\omega \in \mathcal{A}^*$.

Note que operador $\Delta_n(\omega)$ computa a distância entre as probabilidades de transição empíricas para uma sequência ω e a sequência associada $suf(\omega)$.

Exemplo 1.1 (Representação da Árvore de Contextos) *Considere* $\mathbf{X}$ *uma Cadeia de Ordem Variável tomando valores em um alfabeto* $\mathcal{A} = \{0, 1\}$ *e com árvore de contextos* $\mathcal{T}_{\mathbf{X}} = \{0, 01, 11\}$. *Podemos representar* $\mathcal{T}_{\mathbf{X}}$ *da seguinte forma:*

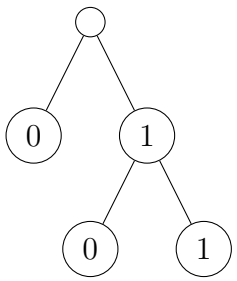


Figura 1.1: Representação da Árvore de Contextos

A profundidade da árvore de contextos é $d(\mathcal{T}_{\mathbf{X}}) = 2$*, pois, os contextos de maior comprimento são* $\omega = 10$ *e* $v = 11$*. A concatenação destes dois contextos é dada por* $\omega v = 1011$*. A sequência* 1 *é o maior sufixo tanto* ω *como* v*, ou seja,* $suf(\omega) = suf(v) = 1$*. A árvore de contextos truncada no nível* $K = 1$ *é dada por* $\mathcal{T}_{\mathbf{X}}|_K = \{0, 1\}$*, que é equivalente a árvore de contextos de uma Cadeia de Markov de ordem* $k = 1$*.*

Exemplo 1.2 (Estimação das Probabilidades de Transição) *Considere* $\mathbf{X}$ *uma Cadeia de Ordem Variável tomando valores num alfabeto* $\mathcal{A} = \{0, 1\}$ *e com árvore de contextos* $\mathcal{T}_{\mathbf{X}} = \{0, 01, 11\}$*. Seja* 111001011010111 *uma amostra aleatória do processo* $\mathbf{X}$*.*

Note que o número de ocorrências das sequências $\omega = 0$*,* $v = 01$ *e* $u = 11$ *foram dadas, respectivamente, por* $N_{15}(\omega) = 5$*,* $N_{15}(v) = 4$ *e* $N_{15}(u) = 5$*. Com intuito de estimar as probabilidades de transição deste processo é necessário determinar o número de ocorrências da concatenação entre cada contexto com o 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* $\hat{p}_X(0|11)_{15} = 0,428$*.*

Capítulo 2

Metodologia

Neste capítulo definimos a versão do Algoritmo Contexto proposta por Galves e Leonardi (2008) utilizada nesse trabalho para a estimação da árvores de contextos. Em seguida, apresentamos os Modelos de Contaminação Estocástica descritos por Collet, Galves e Leonardi (2008) e em Garcia e Moreira (2015). Utilizamos o ambiente R de computação estatística (R Core Team, 2014) para programar o estimador de árvore de contextos e os Modelos de Contaminação Estocástica apresentados neste capítulo.

2.1 Uma versão do Algoritmo Contexto

O algoritmo de estimação da árvore de contextos utilizado nesse trabalho foi proposto por Galves e Leonardi (2008) e é uma modificação do Algoritmo Contexto de Rissanem (1983). Primeiramente, considere o operador $\Delta_n(\omega)$ apresentado na Equação (1.9) do Capítulo 1.

Definição 2.1 (Galves e Leonardi, 2008) *Para todo $\delta > 0$ e $d < n$ a árvore de contextos estimada $\hat{\mathcal{T}}_n^{\delta,d}$ é o conjunto contendo todas as sequências $\omega \in \bigcup_{i=1}^{d} \mathcal{A}_{-i}^{-1}$ tais que $\Delta_n(asuf(\omega)) > \delta$ para algum $a \in \mathcal{A}$ e $\Delta_n(u\omega) \leq \delta$ para todo $u \in \bigcup_{i=1}^{d-l(\omega)} \mathcal{A}_{-i}^{-1}$.*

Note na Definição 2.1 que as constantes $\delta > 0$ e $d < n$ são fundamentais para o estimador, pois, inicialmente é considerada a árvore de contextos maximal. Assim, cada sequência ω candidata a contexto possui comprimento $l(\omega) = d$, ou seja, $\omega \in \mathcal{A}_{-d}^{-1}$. Em seguida, o estimador reduz o comprimento das sequências ω que não satisfazem o critério de poda, apresentado na Definição 2.1, tomando $suf(\omega)$ como novo candidato a contexto. Este procedimento é repetido até que a condição de parada seja satisfeita para todas as sequências $\omega \in \hat{\mathcal{T}}_n^{\delta,d}$.

2.2 Modelos de Contaminação Estocástica

Nesta seção apresentamos três Modelos de Contaminação Estocástica, sendo que um dos modelos foi definido em Collet, Galves e Leonardi (2008) e os outros dois modelos definidos por Garcia e Moreira (2015). Em cada modelo, os autores mostraram que é possível recuperar a árvore de contextos de um processo através de uma amostra contaminada da cadeia segundo um dos modelos de contaminação especificado nos respectivos trabalhos.

Consideramos os processos $\mathbf{X}$ e $\mathbf{Y}$ sendo independentes, não-nulos e com taxa de continuidade somável. Denotamos por $\mathbf{Z} = \{Z_t,\ t \in \mathbb{Z}\}$ os três processos estocasticamente perturbados. A seguir definimos e comentamos os modelos.

Definição 2.2 (Garcia e Moreira, 2015) *Considere* $\mathbf{X} = \{X_t,\ t \in \mathbb{Z}\}$ *um processo estacionário e ergódico tomando valores num alfabeto binário* $\mathcal{A} = \{0, 1\}$. *Seja* $\boldsymbol{\xi} = \{\xi_t, t \in \mathbb{Z}\}$ *uma sequência de variáveis aleatórias Bernoulli i.i.d. tomando valores em* $\{0, 1\}$, *independentes do processo* $\mathbf{X}$, *com*

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

em que ε *é um parâmetro de pertubação fixado em (0,1). Definimos o Modelo de Contaminação Zero Inflado por*

$$Z_t = X_t.\xi_t, \quad t \in \mathbb{Z}. \tag{2.1}$$

Através da Definição 2.2 podemos observar que, no modelo Zero Inflado, a perturbação pode ocorrer apenas quando $X_t = 1$ e $\xi_t = 0$. No entanto, nos modelos apresentados a seguir, em qualquer instante de tempo, a perturbação pode ocorrer para todos dos estados do processo $\mathbf{X}$.

Definição 2.3 (Collet, Galves e Leonardi, 2008) *Considere* $\mathbf{X} = \{X_t,\ t \in \mathbb{Z}\}$ *um processo estacionário e ergódico tomando valores num alfabeto binário* $\mathcal{A} = \{0, 1\}$. *Seja* $\boldsymbol{\xi} = \{\xi_t,\ t \in \mathbb{Z}\}$ *uma sequência de variáveis aleatórias i.i.d. tomando valores em* $\{0, 1\}$, *independentes de* $\mathbf{X}$, *com*

$$\mathbb{P}(\xi_t = 0) = 1 - \varepsilon,$$

em que ε *é um parâmetro de perturbação fixado em* $(0, 1)$. *Definimos o Modelo de Contaminação por Congruência por*

$$Z_t \equiv X_t + \xi_t (mod\ 2), \tag{2.2}$$

em que (2.2) denota a função de congruência módulo 2.

Note na Definição 2.3 que, para o modelo de Contaminação por Congruência, a perturbação pode afetar ambos os estados do processo $\mathbf{X}$. Este modelo

contamina o processo original, sempre que $\xi_t = 1$, trocando o símbolo de X_t. Assim, podemos ver o processo perturbado Z_t como uma função da cadeia original X_t e de ξ_t.

O próximo modelo pode ser resumido do seguinte modo: dadas duas Cadeias de Ordem Variável, tomando valores num mesmo alfabeto finito, a cada instante do tempo, o processo escolhe aleatoriamente um dos dois processos originais com uma probabilidade grande e fixa. A cadeia obtida dessa maneira pode ser vista como uma perturbação estocástica da cadeia que está sendo escolhida com probabilidade maior.

Definição 2.4 (Garcia e Moreira, 2015) *Considere dois processos independentes, estacionários e ergódicos* $\mathbf{X} = \{X_t,\ t \in \mathbb{Z}\}$ *e* $\mathbf{Y} = \{Y_t, t \in \mathbb{Z}\}$ *tomando valores num mesmo alfabeto finito* $\mathcal{A} = \{0, 1, \cdots, N-1\}$*, com* $|\mathcal{A}| = N$*. Seja* $\boldsymbol{\xi} = \{\xi_t,\ t \in \mathbb{Z}\}$ *uma sequência de variáveis aleatórias i.i.d. tomando valores em* $\{0, 1\}$*, independentes dos processos* $\mathbf{X}$ *e* $\mathbf{Y}$*, com*

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

em que ε *é o parâmetro de perturbação fixado em* $(0, 1)$*. Definimos o Modelo de Contaminação por Processo por*

$$Z_t \ = \ \begin{cases} X_t, & \text{se } \xi_t = 1, \\ Y_t, & \text{se } \xi_t = 0. \end{cases} \tag{2.3}$$

Garcia e Moreira (2015) mostraram que o estimador apresentado na Definição 2.1 deste capítulo é robusto, ou seja, se considerarmos os Modelos de Contaminação Zero Inflado ou por Processo apresentados, respectivamente, nas Definições 2.2 e 2.4, mesmo se na estimação utilizarmos uma amostra perturbada do processo, o estimador consegue recuperar a árvore de contextos do processo original. Collet, Galves e Leonardi (2008) mostraram a robustez deste estimador quando consideramos uma amostra perturbada do processo segundo o Modelo de Contaminação por Congruência.

Capítulo 3

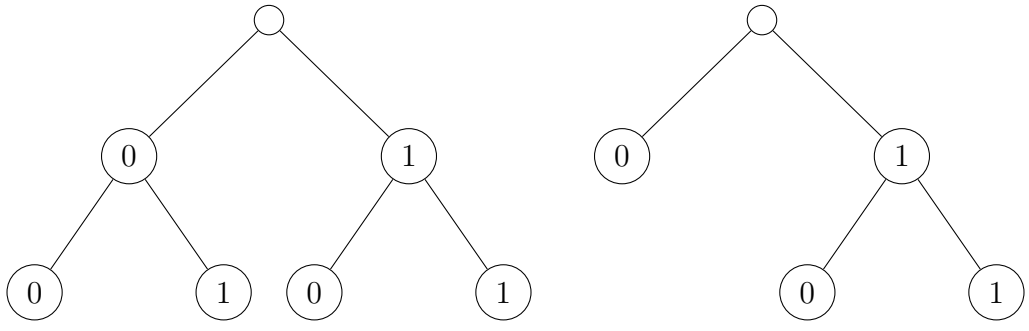
Resultados e Discussão

Neste capítulo avaliamos o desempenho do estimador de árvore de contextos, apresentado na Definição 2.1 do Capítulo 2, em amostras contaminadas segundo os modelos de Contaminação Zero Inflado, por Congruência e por Processo descritos, respectivamente, nas Definições 2.2, 2.3 e 2.4 no Capítulo 1. Em seguida propomos modelos meteorológicos para predizer a possibilidade de o próximo dia ser quente ou não dado as informações de temperaturas máximas dos dias anteriores.

3.1 Simulações de Cadeias de Ordem Variável

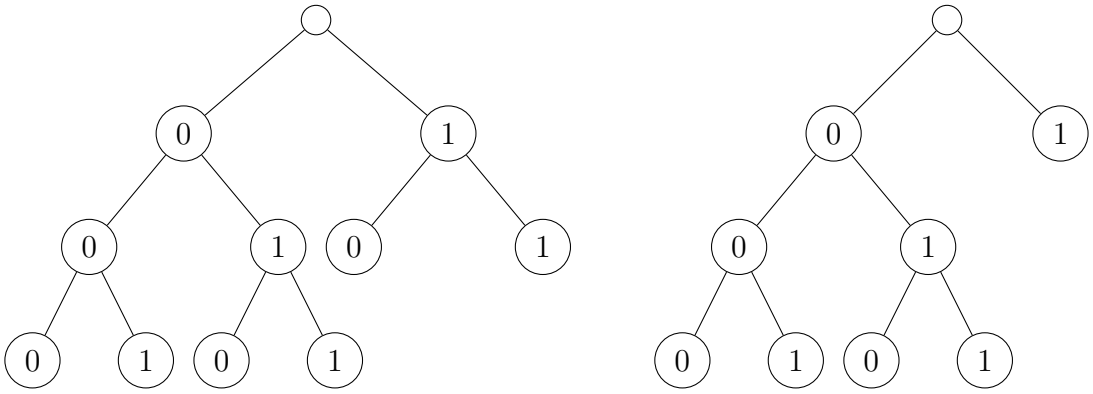
Nesta seção verificamos o comportamento do estimador de árvore de contextos, apresentado na Definição 2.1, utilizando simulações de amostras contaminadas de Cadeias de Ordem Variável.

Seja $\mathbf{X}$ um processo estacionário e ergódico com alfabeto binário $\mathcal{A} = \{0, 1\}$. As Figuras 3.1(a), 3.1(b), 3.2(a), 3.2(b) e 3.3 apresentam as árvores de contextos dos processos simulados.



(a) Árvore 1 (b) Árvore 2

Figura 3.1: Árvores de Contextos 1 e 2



(a) Árvore 3 (b) Árvore 4

Figura 3.2: Árvores de Contextos 3 e 4

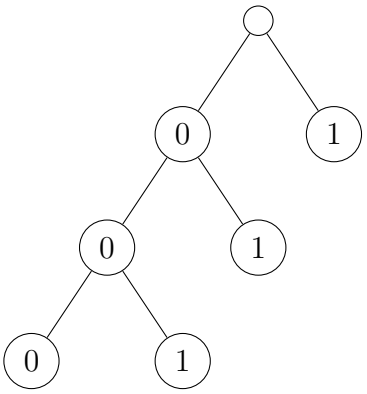


Figura 3.3: Árvore de Contextos 5

Foram realizadas 100 repetições das simulações para cada árvore de contextos, fixado o valor do parâmetro de perturbação ε e para amostras de tamanho $n = 10.000$ e $n = 100.000$. Após as repetições, avaliamos a proporção de retornos corretos da árvore de contextos do processo original.

Os resultados obtidos considerando os Modelos de Contaminação Zero Inflado, por Congruência e por Processo, para tamanho de amostra $n = 10.000$, foram apresentados nas Tabelas 3.1, 3.3 e 3.5, respectivamente. Enquanto os resultados para o tamanho de amosta $n = 100.000$ foram apresentados nas Tabelas 3.2, 3.4 e 3.6.

A Tabela 3.1 apresenta a proporção de retornos corretos da árvore de contextos do processo original, segundo o modelo Zero Inflado, para diferentes valores do parâmetro de perturbação, com tamanho de amostra $n = 10.000$ e 100 repetições. Para as simulações de amostras contaminadas da Árvore 3.1(a), considerando as probabilidades de contaminação fixadas com $\varepsilon \leq 0,01$ a proporção de retorno foi maior ou igual do que o para as simulações da Árvore 3.1(b), sendo que ambas as árvores possuem a mesma profundidade. Entretanto, quando aumentamos o valor do parâmetro de perturbação as simulações da Árvore 3.1(b) apresentaram a maior proporção de retornos

corretos da árvore de contextos do processo original.

Tabela 3.1: Proporção de retornos, modelo de contaminação Zero Inflado, ε fixado, $n = 10.000$ e 100 repetições

Árvore de Contextos	Parâmetro de Perturbação (ε)				
	0,0001	0,001	0,01	0,1	0,20
Árvore 3.1(a)	1,00	1,00	1,00	0,97	0,91
Árvore 3.1(b)	0,99	0,99	1,00	1,00	1,00
Árvore 3.2(a)	0,25	0,34	0,25	0,16	0,27
Árvore 3.2(b)	0,27	0,30	0,23	0,26	0,28
Árvore 3.3	0,29	0,30	0,32	0,64	0,68

Tabela 3.2: Proporção de retornos, modelo de contaminação Zero Inflado, ε fixado, $n = 100.000$ e 100 repetições

Árvore de Contextos	Parâmetro de Perturbação (ε)				
	0,0001	0,001	0,01	0,1	0,20
Árvore 3.1(a)	1,00	1,00	1,00	1,00	0,93
Árvore 3.1(b)	1,00	1,00	1,00	0,99	1,00
Árvore 3.2(a)	0,87	0,89	0,84	0,15	0,17
Árvore 3.2(b)	0,85	0,85	0,86	0,87	0,95
Árvore 3.3	0,86	0,81	0,86	0,99	0,97

Aumentando o tamanho de amostra em cada simulação para $n = 100.000$, considerando ainda o modelo Zero Inflado, podemos observar através da Tabela 3.2 que para as simulações de amostras contaminadas da Árvore 3.1(b), apenas para o valor do parâmetro de perturbação $\varepsilon = 0,1$ a versão do Algoritmo Contexto de Galves e Leonardi (2008) não retornou corretamente a árvore de contextos do processo original em todas as simulações, porém, a proporção de retornos corretos foi de 99%. Por outro lado, para as simulações da Árvore 3.1(a), este estimador da árvore de contextos retornou corretamente em todas as simulações em que $\varepsilon \leq 0,1$ e para $\varepsilon = 0,20$ a proporção de retornos foi de 93%.

Para as simulações de amostras contaminadas das Árvores 3.2(a), 3.2(b) e 3.3, considerando o modelo Zero Inflado, através das Tabelas 3.1 e 3.2 podemos observar que o estimador da árvore de contextos aumentou a proporção de retorno correto a medida que o tamanho de amostra cresceu. Porém, mesmo com $n = 100.000$ o estimador apresentou baixas proporções de retornos corretos para as simulações da Árvore 3.2(a) quando fixamos $\varepsilon \geq 0,10$. Apesar das Árvores 3.2(a), 3.2(b) e 3.3 possuírem a mesma profundidade $d = 3$, as simulações das amostras contaminadas da Árvore 3.3 foi aquela que apresentou o maior retorno correto quando o parâmetro de

perturbação fixado foi $\varepsilon \geq 0,10$. Fixando $\varepsilon \leq 0,01$ as simulações das Árvores 3.2(a), 3.2(b) e 3.3 apresentaram valores próximos da proporção de retorno correto.

As Tabelas 3.3 e 3.4 apresentam a proporção de retornos corretos considerando o modelo de Contaminação por Congruência com o parâmetro de perturbação ε fixado, realizadas 100 repetições, para $n = 10.000$ e $n = 100.000$, respectivamente. As simulações de amostras contaminadas das Árvores 3.1(a) e 3.1(b) com $n = 10.000$, considerando o modelo de contaminação por Congruência, apresentaram alta proporção de retornos corretos com o parâmetro de perturbação fixado $\varepsilon \leq 0,1$. As simulações de amostras contaminadas das Árvores 3.1(a) e 3.1(b) para $\varepsilon = 0,20$ apresentaram baixa proporção de retornos corretos mesmo quando $n = 100.000$.

As simulações de amostras contaminadas das Árvores 3.2(a), 3.2(b) e 3.3, para $n = 10.000$, apresentaram baixas proporções de retornos para todos os valores fixados do parâmetro de perturbação ε. Para o tamanho das amostras $n = 100.000$ as proporções de retornos aumentaram quando fixamos o parâmetro de perturbação $\varepsilon \leq 0,01$, mesmo com esta melhora apenas para $\varepsilon \leq 0,001$ as proporções de retornos foram maiores que 85%.

Tabela 3.3: Proporção de retornos, modelo de contaminação por Congruência, ε fixado, $n = 10.000$ e 100 repetições

Árvore de Contextos	Parâmetro de Perturbação (ε)				
	0,0001	0,001	0,01	0,1	0,20
Árvore 3.1(a)	1,00	1,00	1,00	0,91	0,54
Árvore 3.1(b)	0,99	1,00	0,99	0,86	0,07
Árvore 3.2(a)	0,25	0,24	0,22	0,00	0,00
Árvore 3.2(b)	0,42	0,22	0,16	0,00	0,00
Árvore 3.3	0,28	0,39	0,22	0,00	0,12

Tabela 3.4: Proporção de retornos, modelo de contaminação por Congruência, ε fixado, $n = 100.000$ e 100 repetições

Árvore de Contextos	Parâmetro de Perturbação (ε)				
	0,0001	0,001	0,01	0,1	0,20
Árvore 3.1(a)	1,00	1,00	1,00	0,99	0,37
Árvore 3.1(b)	1,00	1,00	1,00	0,94	0,10
Árvore 3.2(a)	0,86	0,83	0,59	0,00	0,00
Árvore 3.2(b)	0,90	0,87	0,47	0,00	0,00
Árvore 3.3	0,85	0,87	0,29	0,00	0,04

Com intuito de comparação entre os modelos de contaminação definidos no Capítulo 2, consideramos o alfabeto binário $\mathcal{A} = \{0, 1\}$ para o modelo de Contaminação por Processo apresentado na Definição 2.4 do Capítulo 2. Para simularmos amostras

contaminadas segundo este modelo utilizamos a árvore de contextos do processo $\mathbf{Y}$ sendo a apresentada na Figura 3.1(b).

Quando consideramos o modelo de Contaminação por Processo, conforme podemos observar nas Tabelas 3.5 e 3.6, a proporção de retornos aumentou a medida que o tamanho das amostras cresceu de $n = 10.000$ para $n = 100.000$.

Considerando as simulações das Árvores 3.1(a) e 3.1(b), segundo o modelo de Contaminação por Processo e com o valor do parâmetro de perturbação $\varepsilon \leq 0,20$ fixado, conseguimos recuperar a árvore de contexto do processo original com proporção de retornos maior ou igual a 91% para $n = 10.000$. Para as simulações das Árvores 3.2(a), 3.2(b) e 3.3, as proporções de retornos foram baixas para $n = 10.000$, mas aumentaram quando consideramos $n = 100.000$. No entanto, o estimador da árvore de contextos não foi capaz de recuperar a árvore de contextos para $\varepsilon \geq 0.10$.

Tabela 3.5: Proporção de retornos, modelo de contaminação por Processo, ε fixado, $n = 10.000$ e 100 repetições

Árvore de Contextos	Parâmetro de Perturbação (ε)				
	0,0001	0,001	0,01	0,1	0,20
Árvore 3.1(a)	1,00	1,00	1,00	0,99	0,93
Árvore 3.1(b)	0,99	1,00	0,99	0,99	0,91
Árvore 3.2(a)	0,23	0,24	0,21	0,01	0,00
Árvore 3.2(b)	0,17	0,21	0,20	0,00	0,00
Árvore 3.3	0,35	0,25	0,28	0,01	0,00

Tabela 3.6: Proporção de retornos, modelo de contaminação por Processo, ε fixado, $n = 100.000$ e 100 repetições

Árvore de Contextos	Parâmetro de Perturbação (ε)				
	0,0001	0,001	0,01	0,1	0,20
Árvore 3.1(a)	1,00	1,00	1,00	1,00	0,99
Árvore 3.1(b)	1,00	1,00	1,00	1,00	0,93
Árvore 3.2(a)	0,87	0,83	0,71	0,00	0,00
Árvore 3.2(b)	0,91	0,86	0,70	0,00	0,00
Árvore 3.3	0,83	0,88	0,63	0,00	0,00

Através das Tabelas 3.2, 3.4 e 3.6 podemos comparar as proporções de retornos quando consideramos amostras contaminadas segundo os modelos Zero Inflado, por Congruência e por Processo para $n = 100.000$. Quando fixamos alta probabilidade de contaminação, $\varepsilon \geq 0,1$, e perturbamos as amostras segundo o modelo Zero Inflado, para as Árvores 3.2(b) e 3.3, conseguimos recuperar a árvore de contextos do processo original com proporção maior ou igual a 87%. Por outro lado, para os mesmos valores do parâmetro de perturbação, quando perturbamos as amostras segundo os modelos

de Contaminação por Congruência e por Processo o estimador da árvore de contextos não foi capaz de recuperar a árvore do processo original.

Podemos resumir os resultados apresentados nesta seção do seguinte modo: o modelo de Contaminação Zero Inflado apresentou maiores proporções de retornos corretos que os Modelos de Contaminação por Processo e por Congruência, mesmo na presença de alta probabilidade de contaminação. Este fato ocorreu devido a simplicidade do modelo Zero Inflado uma vez que, a cada instante de tempo, o processo original pode ser contaminado, com probabilidade pequena e fixa, apenas se o símbolo do processo nesse instante de tempo for igual a 1. Por outro lado, os modelos de Contaminação por Congruência e por Processo, a cada instante de tempo, podem contaminar a amostra para ambos os símbolos do alfabeto binário, com probabilidade pequena e fixa.

Os modelos de Contaminação por Congruência e por Processo apresentaram desempenhos semelhantes quando comparadas as proporções de retornos corretos da árvore de contexto do processo original, através das simulações de amostras contaminadas segundo estes modelos, com probabilidade de contaminação menor ou igual a $0,001$. No entanto, o modelo de Contaminação por Processo apresentou maiores proporções de retornos quando aumentamos a probabilidade de contaminação para $0,01$.

Destacamos a robustez e o bom comportamento do estimador de árvore de contextos dada uma amostra contaminada do processo. Dessa forma, viabilizando a aplicação destes modelos de contaminação na modelagem de dados meteorológicos apresentados na próxima seção.

3.2 Aplicação de Cadeias de Ordem Variável

Utilizamos dados meteorológicos nos quais constam a medição da temperatura máxima no Distrito Federal para cada dia entre os anos de 2000 a 2014. Os dados podem ser acessados através do portal eletrônico do INMET[1] para a estação Brasília-A001.

Nessa aplicação consideramos o alfabeto binário $\mathcal{A} = \{0, 1\}$. O critério adotado para distinguir entre dias quentes e não quentes foi o seguinte: se a temperatura máxima do dia for maior ou igual ao terceiro quartil da amostra, então o dia é considerado quente. Assim, a cada instante de tempo o processo $\mathbf{X}$ assume 1 se o dia foi quente, ou seja, se a temperatura máxima foi maior ou igual a $28,40°C$, e assume 0 caso contrário. O tamanho da amostra foi de $n = 5.469$.

Para a estimação das árvores de contextos $\mathcal{T}_{\mathbf{X}}$ utilizando a versão do Algoritmo Contexto, apresentada na Definição 2.1 do Capítulo 2, fixamos os parâmetros necessários do estimador $\hat{\mathcal{T}}_n^{\delta,d}$, ou seja, a profundidade das árvores d e o parâmetro

[1] http://www.inmet.gov.br/portal/

$\delta > 0$. Sem perda de generalidade consideramos $d \in \{2, 3, 4\}$. As árvores estimadas são apresentadas nas Figuras 3.4, 3.5 e 3.6.

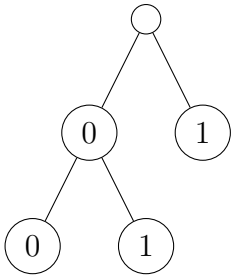


Figura 3.4: Árvore de contextos estimada com profundidade $d = 2$

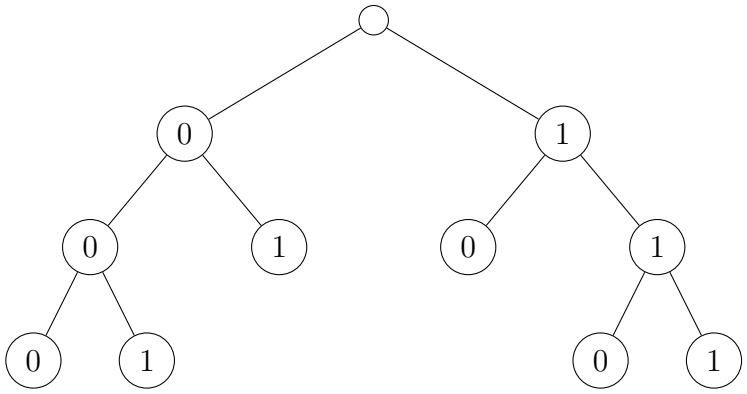


Figura 3.5: Árvore de contextos estimada com profundidade $d = 3$

Com intuito de modelar os dados através de uma Cadeia de Markov de ordem k são necessários $|\mathcal{A}|^k$ parâmetros para predizer o próximo símbolo do processo. Em particular, para $|\mathcal{A}| = 2$ e fixando as ordens k das Cadeias de Markov como sendo 2, 3 ou 4 são necessários, respectivamente, 4, 8 e 16 parâmetros para predizer o próximo símbolo. A ordem destes modelos pode ser estimada, por exemplo, utilizando um dos algoritmos de estimação de ordem descritos em Baigorri, Gonçalves e Resende (2014). Podemos observar nas Figuras 3.4, 3.5 e 3.6 que através da estimação das árvores de contextos com profundidade $d = k$, são necessários apenas 3, 6 ou 12 parâmetros, respectivamente, para d assumindo 2, 3 ou 4.

Uma característica interessante é que em muitas aplicações a forma da árvore de contextos tem uma interpretação natural e informativa. Pela árvore de contextos estimada e apresentada na Figura 3.4, podemos predizer se o próximo dia será quente considerando no máximo as informações dos dois dias anteriores.

Por exemplo, com base na Figura 3.4 a probabilidade de que hoje seja quente dado que ontem foi quente foi de $0,68$. Por outro lado, a probabilidade de que hoje seja quente dado que os dois dias anteriores não foram quentes foi de $0,08$. Se considerarmos que ontem não foi quente e que anteontem foi quente, então a probabilidade estimada

de que o dia presente seja quente foi de $0,27$. Não sendo necessário olhar para mais dias anteriores. Interpretações semelhantes podem ser feitas para as árvores de contextos estimadas e apresentadas nas Figuras 3.5 e 3.6.

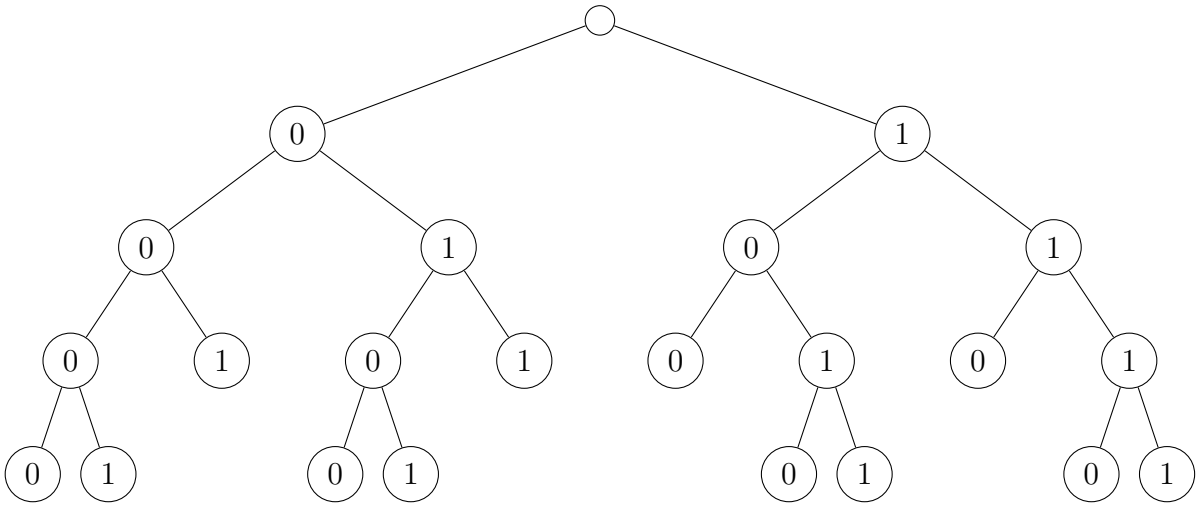


Figura 3.6: Árvore de contextos estimada com profundidade $d = 4$

Para verificar o comportamento do estimador aplicamos uma perturbação à amostra segundo o modelo de Contaminação por Processo, descrito na Definição 2.4 no Capítulo 2. Este modelo de Contaminação é adequado para verificar o comportamento do estimador pois, a cada instante de tempo, a cadeia original pode ser contaminado por outro processo independente, podendo perturbar tando dias quentes como não quentes.

Na perturbação simulamos uma amostra do processo $\mathbf{Y}$ com árvore de contextos dada pela Figura 3.1(b) deste capítulo. A amostra de temperaturas máximas e a amostra simulada do processo $\mathbf{Y}$ são independentes, uma vez que na simulação não foi utilizado nenhuma estrutura de dependência da amostra original.

Desejamos obter, mesmo existindo contaminação, a árvore de contextos do processo original. Garcia e Moreira (2015) provaram que a versão do Algoritmo Contexto, apresentada na Definição 2.1 do Capítulo 2, estima a árvore de contextos de processos, dada uma amostra contaminada segundo o modelo de Contaminação por Processo, desde que a probabilidade de perturbação seja pequena.

Fixando a probabilidade de contaminação $\varepsilon = 0,001$, as árvores de contextos estimada utilizando a amostra contaminada, segundo o modelo de Contaminação por Processo, foi a mesma árvore estimada para a amostra antes de adicionarmos contaminação aos dados. As árvores de contextos estimadas utilizando a amostra contaminada para $\varepsilon = 0,01$ com profundidade $d \in \{2, 3\}$ foram, respectivamente, as mesmas estimadas utilizando a amostra do processo original. Entretanto, para $\varepsilon = 0,01$ e $d = 4$ a árvore de contextos estimada utilizando a amostra contaminada, apesar de não ser a mesma, é similar à árvore estimada através da amostra original. A Figura 3.7 apre-

senta a árvore de contextos estimada utilizando a amostra contaminada com $\varepsilon = 0,01$ e $d = 4$.

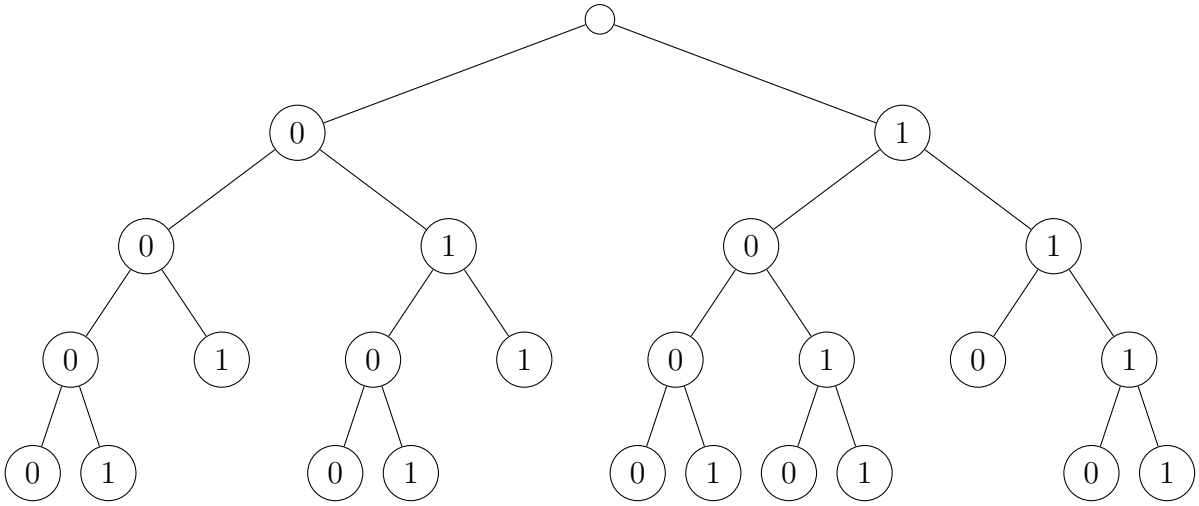


Figura 3.7: Árvore de contextos estimada, segundo o modelo de Contaminação por Processo, com profundidade $d = 4$ e $\varepsilon = 0,01$

Através das Figuras 3.6 e 3.7 podemos observar que utilizando a amostra contaminada, segundo o modelo de Contaminação por Processo com $\varepsilon = 0,01$ e $d = 4$ e pela propriedade do sufixo, conseguimos predizer o próximo símbolo do processo tão bem quanto se utilizarmos a árvore estimada pela amostra original. Assim, se considerarmos a árvore estimada através da amostra contaminada serão necessários 13 parâmetros. No entanto, obtemos uma redução no número de parâmetros necessários em comparação com a modelagem dos dados segundo uma Cadeia de Markov de ordem $k = 4$.

Outra possibilidade para a verificação do comportamento do estimador da árvore de contextos é utilizar o modelo de Contaminação por Congruência, que pode contaminar ambos os possíveis estados do processo com probabilidade suficientemente pequena. Fixando $d \in \{2, 3, 4\}$ e $\varepsilon = 0,001$ as árvores de contextos estimadas foram, respectivamente, as mesmas apresentadas nas Figuras 3.4, 3.5 e 3.6.

Aumentando a probabilidade de contaminação para $\varepsilon = 0,01$, fixando $d \in \{2, 3, 4\}$ e considerando ainda o modelo de Contaminação por Congruência, as árvores estimadas foram distintas das estimadas pela amostra original. Nas Figuras 3.8, 3.9 e 3.10 apresentamos as árvores de contextos estimadas utilizando as amostras contaminadas, segundo este modelo, fixado $d \in \{2, 3, 4\}$ e $\varepsilon = 0,01$.

Através das Figuras 3.8, 3.9 e 3.10, em comparação com a árvore de contextos estimada pela amostra original, para $d = 2$ e $d = 4$ houve aumento no número de parâmetros necessários para predizer o próximo símbolo, enquanto para $d = 3$ houve redução deste número.

Por fim, após perturbarmos aos dados originais, segundo os modelos de

Contaminação por Processo e por Congruência, observamos que quando a versão do Algoritmo Contexto não estimou a mesma árvore de contextos utilizando a amostra original e a contaminada, ainda assim, ambas eram muito semelhantes.

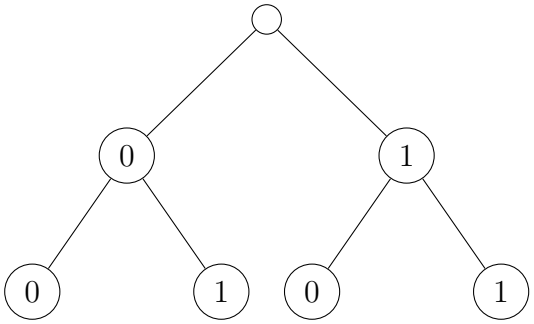


Figura 3.8: Árvore de contextos estimada, segundo o modelo de Contaminação por Congruência, com profundidade $d = 2$ e $\varepsilon = 0,01$

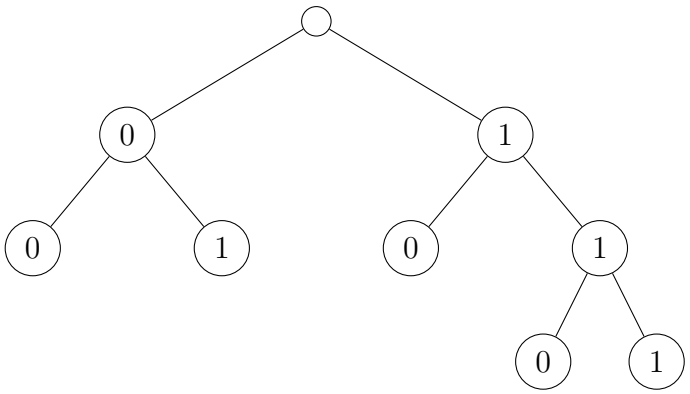


Figura 3.9: Árvore de contextos estimada com profundidade $d = 3$ e $\varepsilon = 0,01$

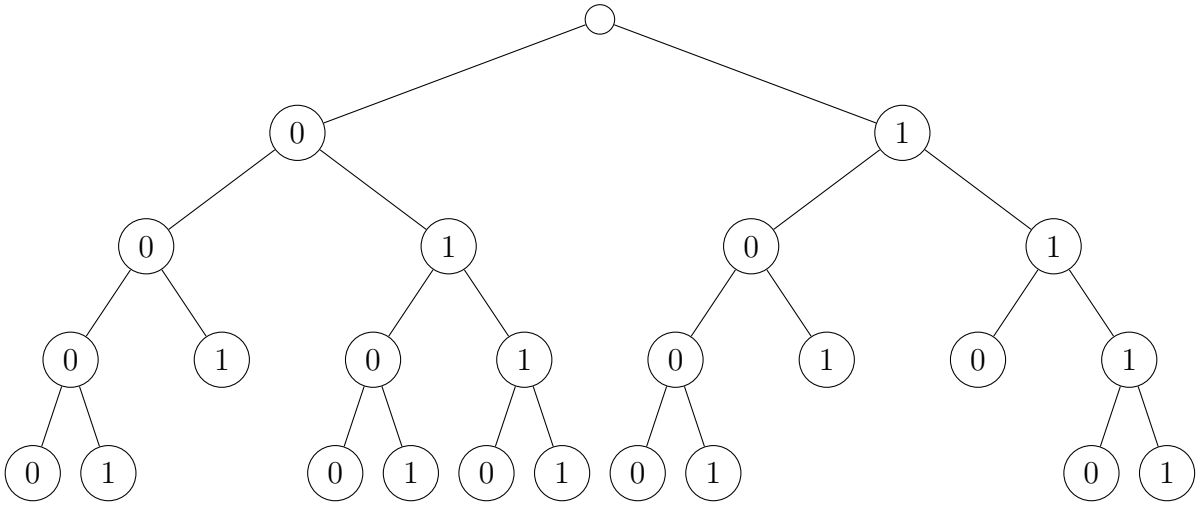


Figura 3.10: Árvore de contextos estimada, segundo o modelo de Contaminação por Congruência, com profundidade $d = 4$ e $\varepsilon = 0,01$

Capítulo 4

Considerações Finais

Neste trabalho estudamos a estimação de árvores de contextos através de uma amostra contaminada do processo original. Na estimação consideramos Cadeias de Ordem Variável tomando valores em um alfabeto binário. Utilizamos os modelos de Contaminação Zero Inflado e por Processo, definidos em Garcia e Moreira (2015), e o modelo de Contaminação por Congruência, definido em Collet, Galves e Leonardi (2008). Em cada um destes modelos de Contaminação os respectivos autores mostraram que é possível recuperar a árvore de contextos do processo original utilizando uma amostra contaminada do processo.

Utilizamos a versão do Algoritmo Contexto apresentada em Galves e Leonardi (2008) para comparação dos modelos de Contaminação. Foram realizadas simulações de amostras perturbadas segundo cada um destes modelos de contaminação e destacamos o bom desempenho do estimador obtido através das proporções de retornos corretos das árvores de contextos do processo original.

Em seguida, como aplicação do modelo de ordem variável modelamos um conjunto de dados meteorológicos. Dessa forma, propomos um modelo para predição de dias quentes e não quentes com base nas informações de temperaturas máximas dos dias anteriores. Destacamos a redução do número de parâmetros necessários para predizer o próximo símbolo do processo em comparação com uma possível modelagem dos dados através de uma Cadeia de Markov de ordem k.

Para verificar o comportamento do estimador adicionamos contaminação aos dados, segundo os modelos de Contaminação por Processo e por Congruência. As árvores estimadas com base nas amostras contaminadas quando não eram as mesmas estimadas pela amostra original, eram muito semelhantes.

Sugerimos para estudos futuros a criação de um estimador para o parâmetro de contaminação que determine se uma amostra está contaminada ou não. Outra sugestão é o aprimoramento dos códigos desenvolvidos em ambiente R, pois, o estimador de árvore de contextos considera qualquer alfabeto finito. No entanto, implementamos este estimador apenas no caso de processos com alfabeto binário.

Referências Bibliográficas

[1] Baigorri, A. R., Gonçalves, C. R., Resende, P. A. A. . Markov chain order estimation based on the chi-square divergence. **Canadian Journal of Statistics**, v. 42, p. 563-578, 2014.

[2] Bühlmann, P., Wyner, A. J. (1999). Variable length Markov chains, **Ann. Statist. 27**: 480-513.

[3] Csiszár, I., Talata, Z. (2006). Context tree estimation for not necessarily finite memory processes, via BIC and MDL, **IEEE Trans. Inform. Theory 52**(3): 1007-1016.

[4] Collet, P., Galves, A., Leonardi, F., Random Perturbations of Stochastic Processes with Unbounded Variable Length Memory. **Eletronic Journal of Probability**, Vol. 13, Paper nº. 48, 13451361,2008.

[5] Duarte, D., Galves, A., Garcia, N. (2006). Markov approximation and consistent estimation of unbounded probabilistic suffix trees, **Bull, Braz. Math. Soc.** p. Aceito.

[6] Ferrari, F. e Wyner, A. (2003). Estimation of general stationary processes by variable length Markov chains, **Scand. J. Statist.**, 30(3): 459-480.

[7] Galves, A., Leonardi, F., Exponential inequalities for empirical unbounded context trees. Vol. 60 of **Progress in Probability**, Birkhauser, 257–270,2008.

[8] Galves, A., Locherbach, E., Stochastic chains with memory of variable length. **TICSP Series 38: 117-133**, 2008.

[9] Galves, A., Maume-Deschamps, V., Schmitt, B., Exponential inequalities for VLMC empirical trees. **ESAIM Prob. Stat.**, 2006.

[10] Garcia, Nancy. L., Moreira, Lucas. Stochastically Perturbed Chains of Variable Memory. **Journal of Statistical Physics**, v. 159, p. 1107-1126, 2015.

[11] Garivier, A., Leonardi, F. Context tree selection: a unifying view. ArXiv:1011.2424v3. **Stochastic processes and their applications** 121, pp. 2488-2506, 2011.

[12] Matta, D. H., **Algoritmos de estimação para Cadeias de Markov de Alcance Variável - aplicações a detecção do ritmo em textos escritos**. Dissertação (Mestrado em Estatística) - Instituto de Matemática, Estatística e Computação Científica, UNICAMP. Campinas, 2008.

[13] 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.

[14] R Core Team (2014). **R: A language and environment for statistical computing**. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

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

Apêndice A

Códigos da versão do Algoritmo Contexto

Neste apêndice apresentamos os códigos desenvolvidos em linguagem R de computação estatística (R Core Team, 2014) da versão do Algoritmo Contexto apresentada na Definição 2.1 do Capítulo 2.

Devemos destacar que durante a implementação do algoritmo optamos pela utilização do alfabeto binário como sendo $\mathcal{A} = \{1, 2\}$, ou seja, a cada elemento de $\{0, 1\}$ foi somado 1.

```
#####################################################
####### Uma versao do  Algoritmo Contexto #########
#####################################################

fcontexto <- function(dados, espaco=2, d, delta, tamanho=length(dados)){

  #Funcao de congruencia ajustada
  congruencia <- function (valor,mod){
    if (valor <= mod) cong <- valor
    if (valor > mod) cong <- ((valor/mod) - floor(valor/mod))*mod
    if (cong == 0) cong<- mod
    cong
  }

  fbase <- function(x,y){ #x=num da linha, y=d
    resto <- 0
    x <- x - 1
    conversao <- (10^y-1)/9
    for (i in y:1){
      resto <- x %% espaco
      conversao <- conversao + resto * 10^(i-1)
      x <- floor(x/espaco)
    }
```

```
  conversao
}

# numero de vezes (funcao indicadora)

vezes <- matrix(0,espaco^(d+1),d+1)

for (comprimento in 1:(d+1)){
  for (tempo in 1:(length(dados)-comprimento+1)){
    i <- 0
    ajuste <- 0
    for (numero in tempo:(tempo+comprimento-1)){
      i <- i + (espaco^ajuste)*(dados[numero]-1)
      ajuste <- ajuste + 1
    }
    vezes[i+1,d+2-comprimento] <- vezes[i+1,d+2-comprimento] + 1
  }
}
for (coluna in 1:d+1){
  for (linha in ((espaco^(d+2-coluna))+1):(espaco^(d+1))){
    vezes[linha,coluna] <- 0
  }
}

# probabilidades de transicao estimadas
transicao <- matrix(0,espaco^(d+1),d+1)

for (coluna in 1:d){
  for (linha in 1:espaco^(d+1)){
    #cont<-ceiling(linha/espaco)
    cont<-congruencia(linha,espaco^(d+1-coluna))
    transicao[linha,coluna] <- (vezes[linha,coluna] + 1)/
    (vezes[cont,coluna+1] + espaco)
    if ((vezes[cont,coluna+1]) == 0) {transicao[linha,coluna] = 0}
  }
}
for(i in 1:espaco){
  transicao[i,d+1] <- (vezes[i,d+1] + 1)/(tamanho + espaco)
}
for (coluna in 1:d+1){
  for (linha in ((espaco^(d+2-coluna))+1):(espaco^(d+1))){
    transicao[linha,coluna] <- 0
  }
}

# operador delta
```

```
operador <- matrix(0,espaco^(d+1),d)

for (coluna in 1:d){
  for (linha in 1:espaco^(d+1)){
    cont=ceiling(linha/espaco)
    operador[linha,coluna] <- transicao[linha,coluna]
    -transicao[cont,coluna+1]
    if (operador[linha,coluna] < 0)
      operador[linha,coluna] <- -operador[linha,coluna]
  }
}

Delta <- matrix(0,espaco^d,d)
for (coluna in 1:d){
  for (linha in 1:((espaco^(d+1-coluna)))){
    vetorcont<-vector("integer",length=espaco)
    for (posicao in 0:(espaco-1)){
      vetorcont[posicao+1] <- operador[posicao*
      (espaco^(d+1-coluna)) + linha, coluna]
      Delta[linha, coluna] <- max(vetorcont)
    }
  }
}

# achando a matriz k
matrizk <- matrix(0,espaco^d,d+1)
for (j in 1:d){
  for (i in 1:espaco^(d-j+1)){
    if (matrizk[i,j] == 1){
      matrizk[ceiling(i/espaco),j+1] <- 1
    }
    else if (matrizk[ceiling(i/espaco),j+1] == 0){
      matrizk[ceiling(i/espaco),j+1] <-
      as.integer(Delta[i,j] > delta)
    }
  }
}

# achando a arvore
arvore <- vector("integer")
arvore[1] <- 0
index <- 1
valor <- 0
for (i in 1:espaco^d){
  for (j in 1:d){
    valor <- 0
```

```
      if (matrizk[i,j] == 0 && matrizk[ceiling(i/espaco),j+1] == 1)
        valor <- 1
      if (valor == 1){
        arvore[index] <- fbase(i,d+1-j)
        index <- index + 1
      }
    }
  }
  arvore
}##Fim da funcao

#########################################################################
################## Utilizacao da funcao  ################################
#########################################################################
#Antes da aplicacao da funcao devem ser definidos:
    ##dados -- vetor do processo com A = {1, 2}
    ##d -- profundidade da arvore a ser estimada
    ##delta -- parametro na Definicao da versao do Algoritmo Contexo

(arvore.estimada <- fcontexto(dados, d, delta))
#########################################################################
```

Apêndice B

Códigos do Algoritmo de Contaminação

Neste apêndice apresentamos os códigos implementados em linguagem R de computação estatística (R Core Team, 2014) do Algoritmo de Contaminação da amostra. Considere neste código o alfabeto binário $\mathcal{A} = \{0, 1\}$.

```
#########################################################
######## Adicionar contaminacao aos dados ##############
#########################################################

contaminacao <- function(dados, modelo, perturb){
  n <- length(dados)

  if(modelo == 1){
    Qsi <- rbinom(n,1,1-perturb)
    #Modelo de Contaminacao Zero Inflado
    Z <- dados*Qsi
  }

  if(modelo == 2){
    Qsi <- rbinom(n,1,perturb)
    #Modelo de Contaminacao por Congruencia
    Z <- ifelse((dados + Qsi) == 1 , 1, 0)
  }

  if(modelo == 3){
    ### Gerando o Processo Y_t
    p <- 0.3
    q <- 0.5
    r <- 0.8

    y <- vector("integer",length = n)
```

```
    y[c(1,2)] <- c(1,0)

    for (i in 3:n){
      v <- runif(1)
      if (y[i-1] == 0){
        if(v < p) y[i] <- 0 else y[i] <- 1}
      if (y[i-2] == 0 && y[i-1] == 1){
        if(v < q) y[i] <- 0 else y[i] <- 1}
      if (y[i-2] == 1 && y[i-1] == 1){
        if(v < r) y[i] <- 0 else y[i] <- 1}
    }
    #Gerar Qsi
    Qsi <- rbinom(n,1,1-perturb)
    #Modelo de Contaminacao por Processo
    Z <- ifelse(Qsi == 1, dados, y)
  }

  Z <- Z + 1

  return(Z)
}## Fim da funcao

#########################################################################
################## Utilizacao da funcao  ################################
#########################################################################
#Antes da aplicacao da funcao devem ser definidos:
    ##dados -- vetor do processo com A = {0, 1}
    ##modelo -- Modelo de Contaminacao
        ####Zero Inflado: modelo = 1
        ####por Congruencia: modelo = 2
        ####por Processo: modelo = 3
    ##perturb -- parametro de perturbacao

(dados.contaminados <- contaminacao(dados, modelo, perturb))
#########################################################################
```