
**Blocos de consenso, esquemas regenerativos e
estimação em tempo polinomial de longas
amostras de cadeias de Markov ocultas**

Suzi Alves Camey

TESE APRESENTADA
AO INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA UNIVERSIDADE DE SÃO PAULO
PARA OBTENÇÃO DO GRAU DE
DOUTOR EM ESTATÍSTICA

Área de Concentração: **Probabilidade e Estatística Aplicada**

Orientador: **Prof. Antonio Galves**

Blocos de consenso, esquemas regenerativos e estimação em tempo polinomial de longas amostras de cadeias de Markov ocultas.

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por Suzi Alves Camey e aprovada pela comissão julgadora.

São Paulo, 08 de agosto de 2005.

Banca Examinadora:

- Prof. Dr. Antonio Galves (Presidente) - IME - USP
- Prof. Dr. Pablo Ferrari - IME - USP
- Prof. Dr. Pierre Collet - CNRS - École Polytechnique
- Prof. Dr. Davide Gabrielli - Università degli Studi di L'Aquila
- Prof. Dr. Ricardo Fraiman - Universidad de San Andres

Dedicatória

Ao meu pai (*in memoriam*)

À Jurema Alcides Cunha (*in memoriam*)

Agradecimentos

Ao meu orientador, Antonio Galves, pelos ensinamentos que vão além do que está contido neste trabalho e pelo apoio nos momentos difíceis.

À Lu, resumidamente: POR TUDO, mas especialmente pelo amor que me dá segurança para continuar perseguindo meus sonhos.

À Luiza, por surgir como uma grande amiga num dos momentos mais difíceis da minha vida.

À Angélica, pelas várias visitas a São Paulo, sempre trazendo chimarrão, e, como ela mesma diz: “por ter me ensinado muiiiiiiiita coisa”.

À Sílvia e à Delhi, pela parceria. Não é sempre que encontramos pessoas que estejam dispostas a partilhar suas vidas conosco.

À minha mãe e ao meu irmão, pelo simples fato de serem minha família. Saber que vocês existem é muito bom.

Aos amigos dos esportes, em especial a Dea e Rodiño. Vocês sabem que são responsáveis pelos momentos mais felizes que vivi em São Paulo.

Aos amigos da minha primeira passagem pelo IME: Gleice, Biss, Dedé e Meiga. Reencontrar vocês e sentir que nossa amizade era a mesma de “alguns” anos atrás me deu a certeza que a distância que nos separa é só um detalhe.

Aos amigos, Sil, Gio e Vânia. A amizade de vocês era uma forte razão para terminar a tese e voltar para Poa.

Aos professores e funcionários do IME, pela disposição em ajudar sempre.

Aos colegas de departamento da UFRGS, em especial a Jandyra, Paty e Flávio, pela torcida, pelo incentivo, pela amizade.

Finalmente, à CAPES e a Universidade Federal do Rio Grande do Sul, pelo auxílio financeiro que possibilitou a execução desta tese.

Resumo

Esta tese propõe duas abordagens para estimar a seqüência oculta de uma cadeia de Markov oculta: blocos de consenso e blocos de regeneração. Em ambos os casos os algoritmos resultantes dependem de um número de operações que cresce polinomialmente com o tamanho da seqüência.

Na primeira abordagem, quebramos a seqüência visível em blocos e estimamos a seqüência oculta de acordo com a maioria de símbolos que enxergamos na seqüência visível. Na segunda abordagem, utilizamos a estrutura regenerativa da cadeia para decompor em blocos independentes. Obtivemos limites superiores para a probabilidade de erro de estimação com os dois métodos.

Na segunda abordagem, utilizamos o método de Monte Carlo markoviano e o algoritmo de Metrópolis para construir iterativamente a seqüência de instantes de regeneração e os blocos correspondentes de estados ocultos, dada a seqüência visível da cadeia.

Na demonstração dos resultados foram utilizados resultados de esquemas regenerativos, o método de Chernoff e a desigualdade de Hoeffding.

Esta tese tem também uma componente computacional. Com efeito, desenvolvemos rotinas em R que implementam os diversos algoritmos propostos.

Também fizemos simulações que ilustram a funcionalidade dos algoritmos.

Abstract

This dissertation introduces two new approaches to the problem of the estimation of the hidden chain in a hidden Markov model. In both approaches the number of steps necessary to estimate the hidden chain increase polynomially with the size of the sample (the observable chain).

In the first approach, through *consensus substrings*, we split the observable chain in substrings and estimate the hidden values as the one which appears in the most of the steps of the observable substring. In the second approach, through *regenerative substrings*, we split the chain using its regenerative structure. In both cases we obtain upper bounds for the error probability of the algorithms.

In the second approach, we use the Monte Carlo Markov chains and the Metropolis algorithm to construct iteratively the sequence of the regeneration times and the corresponding substrings of hidden states, given the observable chain.

The main tools used in the proofs of the theorems were the regenerative construction of the Markov chain, Chernoff's method and Hoeffding's inequality.

This dissertation has also a computational aspect. In effect, all the algo-

rithms introduced here have been implemented in R and the corresponding codes are available in the appendix. We also present several simulations to illustrate the applicability of the algorithms.

Sumário

1	Introdução	1
2	Esquemas regenerativos e cadeias de Markov ocultas	7
2.1	Cadeias de Markov ocultas	7
2.1.1	Métodos para estimação da seqüência oculta	8
2.2	Esquema regenerativo	11
2.3	Construção da cadeia binária regenerativa	16
2.4	Cálculo da verossimilhança usando a estrutura regenerativa . .	21
2.5	Distribuições binomiais associadas a cadeia de Markov oculta .	23
3	Estimação de X_0^T dados os blocos de regeneração	26
3.1	Número de operações para encontrar $\check{X}_0^T(v_0^T, l_0^T)$	29
3.2	Erro médio assintótico de $\check{X}_t(v_0^T, l_0^T)$	34
4	Estimação de X_0^T por blocos de consenso	38
4.1	Algoritmo para encontrar \hat{X}_t	41
4.2	Número de operações para encontrar \hat{X}_0^T	41
4.3	Erro médio assintótico de \hat{X}_t	41
4.4	Estimação da seqüência oculta usando a moda de \hat{X}_t	47

5	Estimação de X_0^T usando método Monte Carlo markoviano	49
5.1	Algoritmo para encontrar \bar{X}_0^T	52
6	Simulações	55
6.1	Simulações para os estimadores $\hat{X}_t(v_0^T, \tilde{l}_0^T)$ e $\hat{X}_t^{(K)}$	55
6.2	Simulações para o estimador $\bar{X}_0^T(v_0^T)$	65
A	Método de Chernoff e desigualdade de Hoeffding	74
B	Rotinas	78
B.1	Programa Xhat.r	78
B.2	Programa Xbarra.r	81
B.3	Programa funcoes2004v2.r	85
B.4	Programa funcoes2004v3.r	91

Lista de Figuras

- 6.1 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,01$ e $\epsilon = 0,01$. 57
- 6.2 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,01$ e $\epsilon = 0,1$. 57
- 6.3 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,01$ e $\epsilon = 0,2$. 58
- 6.4 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,01$ e $\epsilon = 0,3$. 58
- 6.5 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,1$ e $\epsilon = 0,01$. 59
- 6.6 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,1$ e $\epsilon = 0,1$. 59
- 6.7 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,1$ e $\epsilon = 0,2$. 60
- 6.8 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,1$ e $\epsilon = 0,3$. 60
- 6.9 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,2$ e $\epsilon = 0,01$. 61
- 6.10 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,2$ e $\epsilon = 0,1$. 61
- 6.11 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,2$ e $\epsilon = 0,2$. 62
- 6.12 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,2$ e $\epsilon = 0,3$. 62
- 6.13 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,3$ e $\epsilon = 0,01$. 63
- 6.14 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,3$ e $\epsilon = 0,1$. 63
- 6.15 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,3$ e $\epsilon = 0,2$. 64
- 6.16 Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,3$ e $\epsilon = 0,3$. 64
- 6.17 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,01$ e $\epsilon = 0,01$. 66
- 6.18 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,01$ e $\epsilon = 0,1$. 66

-
- 6.19 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,01$ e $\epsilon = 0,2$. 67
- 6.20 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,01$ e $\epsilon = 0,3$. 67
- 6.21 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,1$ e $\epsilon = 0,01$. 68
- 6.22 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,1$ e $\epsilon = 0,1$. 68
- 6.23 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,1$ e $\epsilon = 0,2$. 69
- 6.24 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,1$ e $\epsilon = 0,3$. 69
- 6.25 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,2$ e $\epsilon = 0,01$. 70
- 6.26 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,2$ e $\epsilon = 0,1$. 70
- 6.27 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,2$ e $\epsilon = 0,2$. 71
- 6.28 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,2$ e $\epsilon = 0,3$. 71
- 6.29 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,3$ e $\epsilon = 0,01$. 72
- 6.30 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,3$ e $\epsilon = 0,1$. 72
- 6.31 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,3$ e $\epsilon = 0,2$. 73
- 6.32 Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,3$ e $\epsilon = 0,3$. 73

Capítulo 1

Introdução

Neste trabalho tratamos da identificação dos estados ocultos em longas amostras de cadeias de Markov ocultas. Vamos supor que o alfabeto de valores assumidos pela cadeia oculta e pela visível sejam finitos, conhecidos e coincidam. O nosso objetivo é encontrar procedimentos que envolvam um número de operações que cresce linearmente com o tamanho da amostra.

Vamos considerar dois tipos de abordagem: o de blocos de consenso e dos blocos de regeneração. Ambos são originais e têm como característica comum obter a estimativa global através de uma sucessão de estimativas autônomas em máximos de blocos disjuntos. É isso que permite obter uma estimação com um número de operações proporcional ao tamanho da amostra.

Vamos começar descrevendo os procedimentos de estimação através de blocos de consenso. Vamos supor que a cadeia visível seja obtida alterando com pequena probabilidade ou mantendo com grande probabilidade cada valor da seqüência oculta, procedimento este que é feito de maneira independente em cada ponto. Além disso, vamos supor que a matriz de transição da

cadeia oculta seja muito concentrada na diagonal, isto quer dizer que teremos na seqüência oculta, grandes blocos de símbolos idênticos. Nessas condições, para simplificar a notação, mas sem perda de generalidade, vamos supor que as cadeias sejam binárias. Naturalmente, se a probabilidade de deformação de um ponto for igual a $1/2$ nenhum procedimento de estimação é eficaz. No caso interessante em que a probabilidade de deformação de um ponto for menor que $1/2$, se a matriz de probabilidade de transição da seqüência oculta convergir para a matriz identidade, então a proporção de pontos identificados incorretamente pelo estimador decrescem para zero se o tamanho da amostra crescer mais rapidamente que o comprimento médio dos blocos de coincidência da cadeia oculta.

Vamos agora apresentar informalmente nossos estimadores.

O primeiro estimador utiliza o fato da seqüência oculta possuir grandes blocos de símbolos idênticos. Se a probabilidade de distorção de cada símbolo oculto for pequena, então a seqüência visível é muito semelhante à oculta. Um procedimento natural para estimar a seqüência oculta seria identificar na seqüência visível os pontos onde há mudança na maioria. O ponto delicado desse procedimento é a identificação dos pontos onde ocorre essa mudança.

O procedimento alternativo ao que acabamos de descrever, baseado na mesma idéia da existência de longos blocos ocultos de consenso é quebrar a seqüência de estados visíveis em blocos grandes, mas muito menores que os blocos verdadeiros. Isso faz com que a proporção de blocos visíveis que contém fronteiras de blocos ocultos represente uma fração assintoticamente desprezível do conjunto total de blocos da cadeia visível. Assintótico aqui refere-se ao limite quando a probabilidade de mudança do valor da cadeia

oculta tende a zero. Desta forma os blocos nos quais decomposemos a cadeia visível são grandes e estão tipicamente contidos num bloco maior de consenso da cadeia oculta. Assim, a lei dos grandes números nos assegura que o valor assumido pela maioria dos pontos do bloco visível tipicamente coincide com o valor assumido por todos os pontos da cadeia oculta naquele bloco. As exceções estão nos blocos da cadeia visível que contém pontos de transição da cadeia oculta e a proporção deles é assintoticamente desprezível.

Podemos aumentar a eficácia deste último algoritmo com o procedimento de redivisão e estimação pela moda que descrevemos a seguir. Repetimos várias vezes, independentemente umas das outras, a escolha dos ponto de divisão em blocos da cadeia visível. Em seguida, estimamos o valor da seqüência oculta em cada instante, como sendo aquele que realiza a moda das estimativas daquele ponto nas diversas repetições. A razão pela qual a utilização da moda melhora a estimativa é intuitivamente clara. Mesmo que a chance de deformação de um ponto seja muito pequena, se o tamanho da seqüência for muito longa, teremos muitos blocos para estimar. E é sempre possível termos um bloco no qual a maioria dos pontos tenha um valor deformado. Fixemos um ponto na vizinhança dessa seqüência anômala de pontos deformados. O procedimento de sucessivas escolhas dos pontos de divisão fará com que esse ponto pertença a blocos distintos em cada uma dessas escolhas. Se a probabilidade de um bloco contendo esse ponto conter também uma maioria de pontos deformados for menor que $1/2$, então a estimativa usando a moda nos indicará a resposta justa. O sucesso deste procedimento da moda depende crucialmente da relação entre o tamanho dos blocos visíveis e do comprimento da sucessão de números próximos deformados. O sucesso é garantido

se a probabilidade de deformação for para zero mais rapidamente do que a probabilidade de mudança de maioria.

Vamos agora descrever os procedimentos baseados nos instantes de regeneração da cadeia oculta. É bem conhecido que cadeias de Markov podem ser contruídas através de um esquema regenerativo. Isto significa uma construção acoplada de um processo de renovação e de blocos markovianos independentes correspondendo aos intervalos sucessivos de regeneração (cf. [4]). Vamos supor que conhecemos uma sucessão de instantes de regeneração da cadeia de Markov oculta. Isso nos permite decompor a verossimilhança da cadeia em produto das verossimilhanças de blocos independentes. Essa decomposição nos permite estimar a seqüência oculta com um número de operações que cresce linearmente com o tamanho da amostra.

No caso de não conhecermos os instantes de regeneração da cadeia oculta, mas conhecermos a seqüência visível, e os parâmetros da cadeia, é natural pensar num procedimento iterativo baseado no método de Monte Carlo markoviano com algoritmo de Metrópolis. Isto é feito da seguinte maneira: inicialmente escolhemos ao acaso uma seqüência inicial de candidatos a instantes de regeneração. Dados estes instantes estimamos os estados ocultos. Este procedimento é reiterado de maneira a atualizar os instantes de regeneração e a seqüência oculta, usando o algoritmo de Metrópolis, tendo como critério a comparação das verossimilhanças sucessivas. Este é o último método considerado nesta tese.

Além de resultados teóricos este trabalho tem uma componente computacional. Foram desenvolvidas rotinas em R que implementam os diversos algoritmos propostos. Para mostrar a funcionalidade dos algoritmos foram

realizadas simulações.

Resumindo, nesta tese apresentamos dois tipos de abordagem que acreditamos serem novas para o problema de estimação da seqüência oculta de uma cadeia de Markov oculta: a dos blocos de consenso e a dos blocos regenerativos. Em ambos os casos enunciamos resultados novos que demonstramos rigorosamente e também desenvolvemos programas em R para implementar esses algoritmos (apêndice A). Finalmente apresentamos (capítulo 6) um conjunto de simulações que ilustram os métodos.

Para todos os procedimentos temos que supor que conhecemos os parâmetros da cadeia oculta. No caso dos blocos de consenso a matriz de probabilidades de transição será útil para escolhermos o tamanho médio dos blocos visíveis. No procedimento baseado no método de Monte Carlo, os parâmetros são utilizados no algoritmo de estimação. Para a estimativa do erro cometido pelos estimadores também é necessário o conhecimento dos parâmetros.

Este texto está organizado da seguinte maneira. No capítulo 2 recordamos alguns resultados já conhecidos sobre esquemas regenerativos, cadeias de Markov ocultas e estimação da seqüência oculta em cadeias de Markov ocultas. Em particular, usamos o esquema regenerativo para calcular a verossimilhança da cadeia oculta e encontramos algumas distribuições binomiais associadas à cadeia de Markov oculta. O capítulo 3 é destinado ao estimador que pressupõe o conhecimento dos instantes de regeneração da cadeia oculta. No capítulo 4 é apresentado o estimador por blocos de consenso e finalmente, no capítulo 5, apresentamos o estimador que utiliza o método de Monte Carlo markoviano. No último capítulo são representados resultados de simulação para o estimador por blocos de consenso e para o estimador que utiliza o

método de Monte Carlo markoviano.

Essa tese é parte do Projeto PRONEX/Temático FAPESP *Comportamento Estocástico, Fenômenos Críticos e Identificação de Padrões Rítmicos em Línguas Naturais*. Projeto FAPESP No.03/09930-9.

Capítulo 2

Esquemas regenerativos e cadeias de Markov ocultas

2.1 Cadeias de Markov ocultas

Uma cadeia de Markov oculta é um processo estocástico onde as observações são funções probabilísticas de estados de uma cadeia de Markov, tendo sido apresentadas por Baum e Petrie em 1966 [1]. Esse modelo é um processo estocástico em duas etapas com um processo subjacente não observável (oculto), acessível apenas através de um outro processo estocástico que produz as observações dentro dos estados ocultos. Ou seja, primeiro é gerado um estado, que não vemos; dentro desse estado oculto é gerada uma observação que é a parte visível do processo.

Os elementos que caracterizam uma cadeia de Markov oculta são:

- $A = \{x_1, x_2, \dots, x_N\}$: espaço finito de estados ocultos;

- X_t : estado oculto no instante t ;
- $B = \{v_1, v_2, \dots, v_M\}$: espaço finito de estados observáveis;
- V_t : estado observável no instante t ;
- $Q = \{q(y|x)\}_{x,y \in A}$: matriz de probabilidades de transição da cadeia oculta, onde

$$q(y|x) = P(X_{t+1} = y | X_t = x), \quad x, y \in A; \quad (2.1)$$

- $\mu = \{\mu(x)\}_{x \in A}$: medida estacionária da cadeia oculta;
- $m = \{m_x(v)\}_{x \in A, v \in B}$: distribuição de probabilidades dos estados observáveis dentro de cada estado oculto, ou seja

$$m_x(v) = P(V_t = v | X_t = x), \quad a \in A, v \in B. \quad (2.2)$$

Portanto, para a especificação completa de uma cadeia de Markov oculta devemos conhecer Q e m , que denotaremos somente por θ , isto é, $\theta = (Q, m)$.

2.1.1 Métodos para estimação da seqüência oculta

Dada uma seqüência de observações $V_0^T = V_0, V_1, \dots, V_T$, uma das questões básicas que envolvem o modelo de cadeia de Markov oculta é: como escolher a correspondente seqüência de estados ocultos X_0^T de uma maneira ótima.

O trabalho de Ephraim e Merhav [3] traz uma excelente revisão bibliográfica dos trabalhos desenvolvidos sobre cadeias de Markov ocultas. Nele

podemos encontrar artigos que descrevem alguns algoritmos desenvolvidos para estimar os estados ocultos.

Como visto nos artigos de Khasminskii e Zeitouni [6] e Golubev e Khasminskii [5], podemos medir a performance de um estimador da seqüência oculta através de probabilidade de erro média dada por:

$$D_T^1 \left(\hat{X}_0^T \left(v_0^T, \tilde{l}_0^T \right) \right) = \frac{1}{T} \sum_{t=1}^T \mathbf{1}_{\{\hat{X}_t(v_0^T, \tilde{l}_0^T) \neq X_t\}},$$

onde $\hat{X}_0^T \left(v_0^T, \tilde{l}_0^T \right)$ é o estimador estudado.

O problema que vamos tratar nesta tese é conhecido na literatura como “filtragem”, já que temos $t = T$, isto é, o tamanho da seqüência oculta estimada é igual ao tamanho da seqüência visível observada.

Os critérios mais comuns utilizados para estimar a seqüência oculta X_0^T são a probabilidade mínima de erro do símbolo e probabilidade mínima de erro da seqüência. No primeiro critério é escolhido o estado oculto $\tilde{x}_t \in A$ que minimize a probabilidade de $P(\tilde{X}_t \neq X_t | V_0^T)$, ou seja, a seqüência X_0^T é estimada por \tilde{X}_0^T , onde

$$\tilde{X}_t = \arg \max_{x_t} P_\theta (X_t = x_t | V_0^T = v_0^T),$$

que é uma regra de decisão do símbolo de máximo *a posteriori* (MAP).

De acordo com o segundo critério, teríamos um estimador da seqüência de MAP, ou seja,

$$\hat{X}_0^T = \arg \max_{x_0^T} P_\theta (X_0^T = x_0^T | V_0^T = v_0^T).$$

A seguir mostraremos o algoritmo de Viterbi que segue o segundo critério.

Este algoritmo seleciona passo a passo (a partir do último elemento da seqüência) o estado oculto mais provável, dada a seqüência visível. Ele maximiza o número médio de acertos.

Para descrever o algoritmo precisamos definir:

$$\delta_t(x) = \max_{x_0^{t-1}} P_\theta (X_0^{t-1} = x_0^{t-1}, X_t = x, V_0^t = v_0^t).$$

- Algoritmo de Viterbi

1. Inicialização

$$\delta_0(x) = \mu(x)m_x(v_0), \quad \forall x \in A$$

$$\psi_0(x) = 0, \quad \forall x \in A.$$

2. Indução

$$\delta_t(x) = m_x(v_t) \max_{y \in A} (\delta_{t-1}(y)q(x|y)), \quad \forall x \in A, \quad 1 \leq t \leq T$$

$$\psi_t(x) = \underset{\max_{y \in A}}{\operatorname{argmax}} (\delta_{t-1}(y)q(x|y)) \quad \forall x \in A, \quad 1 \leq t \leq T.$$

3. Finalização

$$P^* = \max_{y \in A} \delta_T(y),$$

$$x_T^* = \underset{\max_{y \in A}}{\operatorname{argmax}} \delta_T(y).$$

4. Seqüência de estados ocultos escolhida

$$x_t^* = \psi_{t+1}(x_{t+1}^*), \quad T-1 \geq t \geq 0.$$

2.2 Esquema regenerativo

A proposição 2.1 nos mostra como podemos relacionar uma cadeia de Markov e uma seqüência de Bernoulli.

Proposição 2.1 *Sejam $(X_t)_{t \geq 0}$ cadeia de Markov com matriz de transição $(q(y|x))_{x,y \in A}$ e $(p^0(y))_{y \in A}$ uma medida de probabilidade qualquer em A . Então podemos construir $(X_t)_{t \geq 0}$ acoplada a uma seqüência $(L_t)_{t \geq 0}$ de variáveis aleatórias independentes e identicamente distribuídas assumindo valores em $\{0, 1\}$ com $\lambda_0 = P(L_t = 0) = 1 - \lambda_1$, de tal forma que:*

$$P(X_{t+1} = y | X_t = x, L_{t+1} = 0) = p^0(y),$$

que independe de x e

$$P(X_{t+1} = y | X_t = x, L_{t+1} = 1) = p^1(y|x),$$

com

$$p^1(y|x) = \frac{q(y|x) - \lambda_0 p^0(y)}{\lambda_1} \quad (2.3)$$

e $\lambda_0 \leq \inf_{x,y \in A} \frac{q(y|x)}{p^0(y)}$.

Prova: Sejam $U_t \sim U(0, 1)$ e $f : (0, 1) \times \{0, 1\} \times A \rightarrow A$. Podemos então definir X_t da seguinte maneira:

$$X_t = f(U_t, L_t, X_{t-1}),$$

com $f(U_t, L_t, X_{t-1}) = y$, se $L_t = 0$ e $\sum_{z \leq y-1} p^0(z) < U_t \leq \sum_{z \leq y} p^0(z)$ ou se $L_t = 1$, $X_{t-1} = x$ e $\sum_{z \leq y-1} p^1(z|x) < U_t \leq \sum_{z \leq y} p^1(z|x)$.

Definindo $(X_t)_{t \geq 0}$ como função do passado e de $(L_t)_{t \geq 0}$ temos que

$$\begin{aligned} P(X_{t+1} = y | X_t = x, L_{t+1} = 0) &= P(f(U_{t+1}, 0, x) = y) \\ &= P\left(\sum_{z \leq y-1} p^0(z) < U_{t+1} \leq \sum_{z \leq y} p^0(z)\right) \\ &= \sum_{z \leq y} p^0(z) - \sum_{z \leq y-1} p^0(z) = p^0(y) \end{aligned}$$

e

$$\begin{aligned} P(X_{t+1} = y | X_t = x, L_{t+1} = 1) &= P(f(U_{t+1}, 1, x) = y) \\ &= P\left(\sum_{z \leq y-1} p^1(z|x) < U_{t+1} \leq \sum_{z \leq y} p^1(z|x)\right) \\ &= \sum_{z \leq y} p^1(z|x) - \sum_{z \leq y-1} p^1(z|x) = p^1(y|x). \end{aligned}$$

Além disso,

$$\begin{aligned} P(X_{t+1} = y | X_t = x) &= P(f(U_{t+1}, 0, x) = y) P(L_{t+1} = 0) \\ &\quad + P(f(U_{t+1}, 1, x) = y) P(L_{t+1} = 1) \\ &= p^0(y)\lambda_0 + p^1(y|x)\lambda_1 = q(y|x). \end{aligned}$$

O que conclui a demonstração. ■

Corolário 2.2 *Sob as mesmas condições da proposição 2.1 temos que*

$$E_{L_t}(p^{L_t}(y|x)) = q(y|x).$$

Prova: Para demonstrar o resultado basta ver que

$$\begin{aligned}
 E_{L_t}(p^{L_t}(y|x)) &= P(L_t = 0)p^0(y) + P(L_t = 1)p^1(y|x) \\
 &= \lambda_0 p^0(y) + \lambda_1 p^1(y|x) \\
 &= q(y|x).
 \end{aligned}$$

■

Corolário 2.3 *Sob as mesmas condições da proposição 2.1, se $\mu(\cdot)$ é a medida estacionária de $q(\cdot|\cdot)$ e $p^0(\cdot) = \mu(\cdot)$, então $\mu(\cdot)$ é a medida estacionária de $p^1(\cdot|\cdot)$.*

Prova: Temos que

$$\begin{aligned}
 \mu(y) &= \sum_x \mu(x)q(y|x) \\
 &= \sum_x \mu(x) [\lambda_0 p^0(y) + \lambda_1 p^1(y|x)] \\
 &= \lambda_0 p^0(y) + \lambda_1 \sum_x \mu(x)p^1(y|x),
 \end{aligned}$$

se tomarmos $p^0(\cdot) = \mu(\cdot)$, temos que

$$\begin{aligned}
 \mu(y) &= \frac{\lambda_1}{(1 - \lambda_0)} \sum_x \mu(x)p^1(y|x) \\
 &= \sum_x \mu(x)p^1(y|x).
 \end{aligned}$$

Logo, $\mu(\cdot)$ também é medida estacionária de $p^1(\cdot|\cdot)$. ■

Para o caso $|A| = 2$ a forma explícita de $(p^0(y))_{y \in A}$ e $(p^1(y|x))_{x,y \in A}$ vai ser apresentada na proposição abaixo.

Proposição 2.4 *Sejam $(X_t)_{t \geq 0}$ uma cadeia de Markov com matriz de transição $(q(y|x))_{x,y \in A}$ e $|A| = 2$, se tomarmos*

$$\lambda_0 = \sum_{y \in A} \min_{x \in A} q(y|x) \quad (2.4)$$

e

$$p^0(y) = \frac{\min_{x \in A} q(y|x)}{\lambda_0}, \quad \forall y \in A \quad (2.5)$$

temos que:

- se $q(x|x) > q(x|y)$ então $p^1(x|x) = 1$;
- se $q(x|x) < q(x|y)$ então $p^1(x|x) = 0$.

Prova: Vamos provar somente a primeira parte, a segunda é análoga.

Se $q(x|x) > q(x|y)$ então $q(y|x) < q(y|y)$, logo

$$\lambda_0 = q(x|y) + q(y|x)$$

e

$$p^0(x) = \frac{q(x|y)}{\lambda_0}.$$

Deste modo, por (2.3) vemos que

$$\begin{aligned} p^1(x|x) &= \frac{q(x|x) - \lambda_0 p^0(x)}{1 - (q(x|y) + q(y|x))} \\ &= \frac{q(x|x) - q(x|y)}{q(x|x) - q(x|y)} = 1 \\ &= p^1(y|y). \end{aligned}$$

E,

$$\begin{aligned} p^1(x|y) &= \frac{q(x|y) - \lambda_0 p^0(y)}{1 - (q(x|y) + q(y|x))} \\ &= \frac{q(x|y) - q(x|y)}{q(x|x) - q(x|y)} = 0 \\ &= p^1(y|x). \end{aligned}$$

■

Quando a matriz de transição $(q(y|x))_{x,y \in A}$ é simétrica temos um interessante resultado que será enunciado no corolário 2.5.

Corolário 2.5 *Sejam $(X_t)_{t \geq 0}$ uma cadeia de Markov com matriz de transição $(q(y|x))_{x,y \in A}$ e $|A| = 2$, com $q(y|x) = q(x|y) = q$, se tomarmos λ_0 e $p^0(y)$ como em 2.4 e 2.5, respectivamente, temos que $p^0(\cdot) = 1/2$ e*

- se $q < 1/2$, então $\lambda_0 = 2q$;
- se $q > 1/2$, então $\lambda_0 = 2(1 - q)$.

Prova: Se $q < 1/2$, então $q < 1 - q$ e portanto

$$\min_{x \in A} q(y|x) = q, \quad \forall y,$$

logo, $\lambda_0 = 2q$.

A prova é análoga para $q > 1/2$. E, em ambos os casos,

$$p^0(y) = \frac{q}{2q} = \frac{1 - q}{2(1 - q)} = \frac{1}{2}.$$

O que conclui a demonstração. ■

2.3 Construção da cadeia binária regenerativa

Neste capítulo vamos definir como se constrói uma cadeia de Markov oculta com espaço de estados $A = B = \{0, 1\}$, matriz de transição

$$\mathbf{Q} = \begin{pmatrix} 1 - q & q \\ q & 1 - q \end{pmatrix},$$

com $q < 1/2$, e

$$\mathbf{m} = \begin{pmatrix} 1 - \epsilon & \epsilon \\ \epsilon & 1 - \epsilon \end{pmatrix},$$

com $\epsilon < 1/2$, usando tempos de regeneração.

Primeiro vamos introduzir a seqüência de variáveis aleatórias que definem os instantes de regeneração. Seja $(L_t)_{t \geq 1}$ uma seqüência de variáveis aleatórias independentes e identicamente distribuídas assumindo valores em $\{0, 1\}$ com

$$P(L_t = 0) = 2q \text{ e } P(L_t = 1) = 1 - 2q, \text{ com } q < 1/2. \quad (2.6)$$

Os instantes de regeneração associados a seqüência $(L_t)_{t \geq 1}$ ficam assim definidos:

$$\tau_0 = 0,$$

$$\tau_i = \min\{t > \tau_{i-1} : L_t = 0\}, \quad i \geq 1. \quad (2.7)$$

Para todo $T \geq 1$, definimos o número de regenerações, J_T , por:

$$J_T = \sum_{t=1}^T \mathbf{1}_{\{L_t=0\}}. \quad (2.8)$$

Vamos introduzir outra seqüência de variáveis aleatórias que será utilizada na construção da cadeia de Markov oculta. Seja $(\xi_t)_{t \geq 0}$ uma seqüência de variáveis aleatórias independentes e identicamente distribuídas assumindo valores em $\{0, 1\}$ com

$$P(\xi_t = 0) = 1/2 = P(\xi_t = 1). \quad (2.9)$$

Utilizando as duas seqüências introduzidas anteriormente, temos que a seqüência oculta $(X_t)_{t \geq 0}$ é definida da seguinte maneira:

Definição 2.6 *Seja $X_t \in \{0, 1\}$, $t \geq 0$ definida por:*

$$X_t = \xi_i, \text{ se } \tau_i \leq t < \tau_{i+1},$$

com τ_i e $(\xi_t)_{t \geq 0}$ como em (2.7) e (2.9), respectivamente.

Para definir a seqüência visível, precisamos introduzir outra seqüência de variáveis aleatórias que chamaremos de $(\delta_t)_{t \geq 0}$.

Seja $(\delta_t)_{t \geq 0}$ uma seqüência de variáveis aleatórias independentes e identicamente distribuídas assumindo valores em $\{0, 1\}$ com

$$P(\delta_t = 0) = \epsilon \text{ e } P(\delta_t = 1) = 1 - \epsilon, \quad (2.10)$$

com $\epsilon < 1/2$. Agora podemos definir a seqüência de variáveis visíveis $(V_t)_{t \geq 0}$.

Definição 2.7 *Seja $V_t \in \{0, 1\}$, $t \geq 0$ definida por:*

$$V_t = \delta_t X_t + (1 - \delta_t)(1 - X_t), \quad \forall t \geq 0,$$

com $(X_t)_{t \geq 0}$ de acordo com a definição 2.6 e $(\delta_t)_{t \geq 0}$ como em (2.10).

A proposição 2.8 mostra a matriz de transição da cadeia de Markov que acabamos de construir e a relação entre $(X_t)_{t \geq 0}$ e $(L_t)_{t \geq 1}$.

Proposição 2.8 *Sejam $(L_t)_{t \geq 1}$ como em (2.6) e $(X_t)_{t \geq 0}$ conforme definição 2.6. Então, a matriz de transição de $(X_t)_{t \geq 0}$ é*

$$\mathbf{Q} = \begin{pmatrix} 1 - q & q \\ q & 1 - q \end{pmatrix},$$

com $q < 1/2$, e $\forall x, y \in \{0, 1\}$,

$$P(X_{t+1} = y | X_t = x, L_{t+1} = 0) = \frac{1}{2},$$

que independe de x e

$$P(X_{t+1} = y | X_t = x, L_{t+1} = 1) = \mathbf{1}_{\{x=y\}}.$$

Prova: Sabemos que existe $i \in \{0, 1, \dots, J_T\}$ tal que $\tau_i \leq t < \tau_{i+1} \leq T$, assim temos que:

$$[X_{t+1} = x | X_t = x] = [L_{t+1} = 1] \cup [L_{t+1} = 0, \xi_{t+1} = x],$$

logo,

$$\begin{aligned}
 P(X_{t+1} = x | X_t = x) &= P(L_{t+1} = 1 \cup L_{t+1} = 0, \xi_{t+1} = x) \\
 &= P(L_{t+1} = 1) + P(L_{t+1} = 0, \xi_{t+1} = x) \\
 &= 1 - 2q + 2q \frac{1}{2} = 1 - q.
 \end{aligned}$$

Da mesma forma,

$$[X_{t+1} = y | X_t = x] = [L_{t+1} = 0, \xi_{t+1} = y],$$

e conseqüentemente

$$\begin{aligned}
 P(X_{t+1} = y | X_t = x) &= P(L_{t+1} = 0, \xi_{t+1} = y) \\
 &= 2q \frac{1}{2} \\
 &= q.
 \end{aligned}$$

E portanto

$$\mathbf{Q} = \begin{pmatrix} 1 - q & q \\ q & 1 - q \end{pmatrix},$$

com $q < 1/2$.

Além disso, se $L_{t+1} = 0$ então $\tau_i \leq t < t + 1 = \tau_{i+1}$ e portanto

$$P(X_{t+1} = y | X_t = x, L_{t+1} = 0) = P(\xi_{i+1} = y) = \frac{1}{2}$$

e se $L_{t+1} = 1$ então $\tau_i \leq t < t + 1 < \tau_{i+1}$, que implica em $X_{t+1} = X_t$, logo

$$P(X_{t+1} = y | X_t = x, L_{t+1} = 1) = \begin{cases} 1 & \text{se } x = y \\ 0 & \text{se } x \neq y \end{cases},$$

o que termina de demonstrar a proposição. ■

A proposição 2.9 mostra que

$$\mathbf{m} = \begin{pmatrix} 1 - \epsilon & \epsilon \\ \epsilon & 1 - \epsilon \end{pmatrix},$$

com $\epsilon < 1/2$.

Proposição 2.9 *Sejam $(X_t)_{t \geq 0}$ e $(V_t)_{t \geq 0}$ como nas definições 2.6 e 2.7, respectivamente. Então*

$$\mathbf{m} = \begin{pmatrix} 1 - \epsilon & \epsilon \\ \epsilon & 1 - \epsilon \end{pmatrix},$$

com $\epsilon < 1/2$.

Prova: Pela definição 2.7 temos que

$$\begin{aligned} P(V_t = v | X_t = x) &= P(\delta_t X_t + (1 - \delta_t)(1 - X_t) = v | X_t = x) \\ &= P(\delta_t x + (1 - \delta_t)(1 - x) = v | X_t = x) \\ &= P\left(\delta_t = \frac{1 - x - v}{1 - 2x}\right), \end{aligned}$$

2.4 Cálculo da verossimilhança usando a estrutura regenerativa 21

Assim, lembrando que $v \neq x$ implica que $v = 1 - x$, temos que

$$P(V_t = v | X_t = x) = \begin{cases} P(\delta_t = 1) = 1 - \epsilon & \text{se } v = x \\ P(\delta_t = 0) = \epsilon & \text{se } v \neq x \end{cases},$$

com $\epsilon < 1/2$. O que conclui a demonstração. ■

2.4 Cálculo da verossimilhança usando a estrutura regenerativa

Uma importante aplicação da decomposição em blocos regenerativos independentes é o cálculo da verossimilhança da cadeia oculta, como produto das verossimilhanças dos blocos independentes.

Proposição 2.10 *Sejam $(X_t)_{t \geq 0}$ e $(V_t)_{t \geq 0}$ conforme as definições 2.6 e 2.7. Sejam $(L_t)_{t \geq 0}$, τ_i e J_T como em (2.6), (2.7) e (2.8) respectivamente, então*

$$P_\theta(X_0^T = x_0^T | V_0^T = v_0^T, L_0^T = l_0^T) = B_{início} * \prod_{i=1}^{J_T-1} B_i * B_{fim}, \quad (2.11)$$

onde

$$\begin{aligned} B_{início} &= P_\theta(X_0^{\tau_1-1} = x_0^{\tau_1-1} | V_0^{\tau_1-1} = v_0^{\tau_1-1}, L_0^{\tau_1-1} = l_0^{\tau_1-1}), \\ B_i &= P_\theta(X_{\tau_i}^{\tau_{i+1}-1} = x_{\tau_i}^{\tau_{i+1}-1} | V_{\tau_i}^{\tau_{i+1}-1} = v_{\tau_i}^{\tau_{i+1}-1}, L_{\tau_i}^{\tau_{i+1}-1} = l_{\tau_i}^{\tau_{i+1}-1}), \\ B_{fim} &= P_\theta(X_{\tau_{J_T}}^T = x_{\tau_{J_T}}^T | V_{\tau_{J_T}}^T = v_{\tau_{J_T}}^T, L_{\tau_{J_T}}^T = l_{\tau_{J_T}}^T). \end{aligned}$$

2.4 Cálculo da verossimilhança usando a estrutura regenerativa 22

Prova: Por definição

$$P_\theta (X_0^T = x_0^T | V_0^T = v_0^T, L_0^T = l_0^T)$$

é igual a

$$\frac{P_\theta (X_0^T = x_0^T, V_0^T = v_0^T, L_0^T = l_0^T)}{\sum_{y_0^T \in A^T} P_\theta (X_0^T = y_0^T, V_0^T = v_0^T, L_0^T = l_0^T)}, \quad (2.12)$$

mas, pelas definições de $(L_t)_{t \geq 0}$, $(X_t)_{t \geq 0}$ e $(V_t)_{t \geq 0}$, temos que

$$\begin{aligned} P_\theta (X_0^T = x_0^T, V_0^T = v_0^T, L_0^T = l_0^T) &= P_\theta (V_0^T = v_0^T | X_0^T = x_0^T, L_0^T = l_0^T) P_\theta (X_0^T = x_0^T | L_0^T = l_0^T) P_\theta (L_0^T = l_0^T) \\ &= \lambda_0^{J_T} (1 - \lambda_0)^{T - J_T} \mu(x_0) m_{x_0}(v_0) \prod_{t=1}^T p^{lt}(x_t | x_{t-1}) m_{x_t}(v_t), \end{aligned}$$

logo,

$$P_\theta (X_0^T = x_0^T | V_0^T = v_0^T, L_0^T = l_0^T) = \frac{\mu(x_0) m_{x_0}(v_0) \prod_{t=1}^T p^{lt}(x_t | x_{t-1}) m_{x_t}(v_t)}{\sum_{y_0^T \in A^T} \mu(y_0) m_{y_0}(v_0) \prod_{t=1}^T p^{lt}(y_t | y_{t-1}) m_{y_t}(v_t)}.$$

Mas, que pela independência dos blocos, pode ser escrito como

$$P_\theta (X_0^T = x_0^T | V_0^T = v_0^T, L_0^T = l_0^T) = B_{início} * \prod_{i=1}^{J_T - 1} B_i * B_{fim},$$

como queríamos demonstrar. ■

2.5 Distribuições binomiais associadas a cadeia de Markov oculta 23

O corolário a seguir prova que sob as seguintes condições: $A = B = \{0, 1\}$, $q(x|x) > q(x|y)$ e $m_x(x) = 1 - \epsilon$, todos os elementos de um mesmo bloco são iguais a zero ou iguais a um.

Corolário 2.11 *Se $A = B = \{0, 1\}$, $q(x|x) > q(x|y)$ e $m_x(x) = 1 - \epsilon$, então*

$$P_\theta (X_{\tau_i}^{\tau_{i+1}-1} = x_{\tau_i}^{\tau_{i+1}-1}, V_{\tau_i}^{\tau_{i+1}-1} = v_{\tau_i}^{\tau_{i+1}-1}, L_{\tau_i}^{\tau_{i+1}-1} = l_{\tau_i}^{\tau_{i+1}-1})$$

é igual a

$$\begin{cases} \lambda_0 \lambda_1^{l_i} p^0(x_{\tau_i}) (1 - \epsilon)^{N_i} \epsilon^{l_i - N_i}, & \text{se } x_s = x_{\tau_i}, \tau_i \leq s < \tau_{i+1}, \\ 0, & \text{caso contrário} \end{cases},$$

onde $N_i = \sum_{s=\tau_i}^{\tau_{i+1}-1} \mathbf{1}\{v_s = x_{\tau_i}\}$ e $l_i = \tau_i - \tau_{i-1}$.

Prova: Pela proposição 2.4, se $q(x|x) > q(x|y)$ temos que $p^1(x|x) = 1$ e $p^1(y|x) = 0$. Aplicando este resultado à proposição 2.10 concluímos a demonstração. ■

2.5 Distribuições binomiais associadas a cadeia de Markov oculta

Nesta seção enunciaremos alguns lemas que trazem algumas distribuições que serão úteis para o nosso trabalho. Será considerado aqui o seguinte caso $A = B = \{0, 1\}$, $q(x|x) = 1 - q$, $q(1 - x|x) = q$, $q < 1/2$, $m_x(x) = 1 - \epsilon$, $m_x(1 - x) = \epsilon \forall x \in A$ e $\epsilon < 1/2$.

2.5 Distribuições binomiais associadas a cadeia de Markov oculta 24

Lema 2.12 *Seja $(X_t)_{t \geq 0}$, conforme definição 2.6, então*

$$\sum_{s=s_1}^{s_2-1} \mathbf{1}_{\{X_s \neq X_{s+1}\}} \sim \beta(s_2 - s_1; p).$$

Prova: Vamos começar provando que $[X_s \neq X_{s+1}]$ e $[X_{s+1} \neq X_{s+2}]$ são independentes. Como $P_\theta(X_s \neq X_{s+1}) = p$ temos que

$$P_\theta(X_s \neq X_{s+1}) P_\theta(X_{s+1} \neq X_{s+2}) = p^2,$$

assim para provar a independência só precisamos mostrar que

$$P_\theta(X_s \neq X_{s+1}, X_{s+1} \neq X_{s+2}) = p^2.$$

Entretanto,

$$\begin{aligned} P_\theta(X_s \neq X_{s+1}, X_{s+1} \neq X_{s+2}) &= \sum_{x=0,1} P_\theta(X_s = x, X_{s+1} = 1 - x, X_{s+2} = x) \\ &= \sum_{x=0,1} \frac{1}{2} p^2 \\ &= p^2. \end{aligned}$$

Logo,

$$\sum_{s=s_1}^{s_2-1} \mathbf{1}_{\{X_s \neq X_{s+1}\}}$$

é uma soma de variáveis aleatórias binárias independentes com $P_\theta(X_s \neq X_{s+1}) = p$, isto é, é uma binomial com parâmetros $s_2 - s_1$ e p . ■

Lema 2.13 *Sejam $(X_t)_{t \geq 0}$ e $(V_t)_{t \geq 0}$, conforme definições 2.6 e 2.7, respec-*

2.5 Distribuições binomiais associadas a cadeia de Markov oculta 25

tivamente, então dado que $X_{s_1}^{s_2} = x$

$$\sum_{s=s_1}^{s_2-1} \mathbf{1}_{\{V_s=1-x\}} \sim \beta(s_2 - s_1; \epsilon).$$

Prova: A prova é imediata, pois, por definição, dado $X_{s_1}^{s_2}$ as variáveis aleatórias V_s , $s \in [s_1; s_2)$ são independentes e também $P_\theta(V_s = 1 - x | X_s = x) = m_x(1 - x) = \epsilon$, logo dado que $X_{s_1}^{s_2} = x$

$$\sum_{s=s_1}^{s_2-1} \mathbf{1}_{\{V_s=1-x\}}$$

é uma binomial com parâmetros $s_2 - s_1$ e ϵ . ■

Capítulo 3

Estimação de X_0^T dados os blocos de regeneração

O primeiro estimador proposto é $\ddot{X}_0^T(v_0^T, l_0^T)$ definido por:

Definição 3.1 *Sejam $v_0^T \in B^T$ e $l_0^T \in \{0, 1\}^T$ conhecidos. Então*

$$\ddot{X}_0^T(v_0^T, l_0^T) = \underset{x_0^T}{\operatorname{argmax}} P_\theta(X_0^T = x_0^T | V_0^T = v_0^T, L_0^T = l_0^T). \quad (3.1)$$

Caso exista mais de uma seqüência que maximiza a verossimilhança então escolhemos ao acaso uma delas.

A proposição 3.2 mostra que este estimador é obtido através dos blocos independentes.

Proposição 3.2 *Sob as mesmas condições da proposição 2.10 temos que*

$$\ddot{X}_0^T(v_0^T, l_0^T) = (\ddot{X}_{\tau_j}^{\tau_{j+1}-1})_{j=0, \dots, J_T},$$

onde

$$\ddot{X}_{\tau_j}^{\tau_{j+1}-1} = \underset{x_{\tau_j}^{\tau_{j+1}-1}}{\operatorname{argmax}} P_\theta \left(X_{\tau_j}^{\tau_{j+1}-1} = x_{\tau_j}^{\tau_{j+1}-1}, V_{\tau_j}^{\tau_{j+1}-1} = v_{\tau_j}^{\tau_{j+1}-1}, L_{\tau_j}^{\tau_{j+1}-1} = l_{\tau_j}^{\tau_{j+1}-1} \right).$$

Prova: Pela proposição 2.10

$$\underset{x_0^T}{\operatorname{argmax}} P_\theta \left(X_0^T = x_0^T | V_0^T = v_0^T, L_0^T = l_0^T \right)$$

é a seqüência dos argumentos do máximo de cada um dos termos do produto de (2.11). E por (2.12), temos que

$$\underset{x_i^j}{\operatorname{argmax}} P_\theta \left(X_i^j = x_i^j | V_i^j = v_i^j, L_i^j = l_i^j \right) = \underset{x_i^j}{\operatorname{argmax}} P_\theta \left(X_i^j = x_i^j, V_i^j = v_i^j, L_i^j = l_i^j \right).$$

Donde se conclui a prova. ■

Pelo corolário 2.11 a escolha do argumento do máximo restringe-se às seqüências onde todos estados são iguais a 0 ou iguais a 1. O lema a seguir nos mostra quando o argumento do máximo dentro de um bloco é uma seqüência de zeros ou de uns.

Lema 3.3 *Se $A = B = \{0, 1\}$, $q(x|x) = q(y|y) = q < 1/2$ e $m_x(v) = 1 - \epsilon$ se $x = v$, então para todo $t \in [\tau_{i-1}^t; \tau_i^t)$*

$$\ddot{X}_t(v_0^T, l_0^T) = \begin{cases} 0, & \text{se } N_0 > \frac{l_i(t)}{2} \\ 1, & \text{se } N_1 > \frac{l_i(t)}{2} \\ \mathbf{1}_{\{U \leq 1/2\}}, & \text{se } N_0 = N_1 = \frac{l_i}{2} \end{cases},$$

onde $\tau_{i-1}^t = \max\{s \leq t : L_s = 0\}$, $\tau_i^t = \min\{s > t : L_s = 0\}$, $l_i(t) = \tau_i^t - \tau_{i-1}^t$, $N_x = \sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} \mathbf{1}_{\{v_s=x\}}$ e $U \sim \text{Uniforme}[0, 1]$.

Prova: Pelo corolário 2.11, como $A = B = \{0, 1\}$, $q(x|x) > q(x|y)$ e $m_x(x) = 1 - \epsilon$, então

$$\begin{aligned} \frac{P_\theta \left(X_{\tau_{i-1}^t}^{\tau_i^t-1} = \mathbf{0}, V_{\tau_{i-1}^t}^{\tau_i^t-1} = v_{\tau_{i-1}^t}^{\tau_i^t-1}, L_{\tau_{i-1}^t}^{\tau_i^t-1} = l_{\tau_{i-1}^t}^{\tau_i^t-1} \right)}{P_\theta \left(X_{\tau_{i-1}^t}^{\tau_i^t-1} = \mathbf{1}, V_{\tau_{i-1}^t}^{\tau_i^t-1} = v_{\tau_{i-1}^t}^{\tau_i^t-1}, L_{\tau_{i-1}^t}^{\tau_i^t-1} = l_{\tau_{i-1}^t}^{\tau_i^t-1} \right)} &= \frac{\lambda_0 \lambda_1^{l_i(t)} (1/2) (1 - \epsilon)^{N_0} \epsilon^{l_i(t) - N_0}}{\lambda_0 \lambda_1^{l_i(t)} (1/2) (1 - \epsilon)^{N_1} \epsilon^{l_i(t) - N_1}} \\ &= \frac{(1 - \epsilon)^{N_0} \epsilon^{l_i(t) - N_0}}{(1 - \epsilon)^{N_1} \epsilon^{l_i(t) - N_1}} \\ &= \left(\frac{1 - \epsilon}{\epsilon} \right)^{N_0 - N_1}, \end{aligned}$$

pois como $q(x|x) = q(y|y) = q$, pelo corolário 2.5, $p^0(0) = p^0(1) = 1/2$.

Logo, se $N_0 > N_1$, então

$$\left(\frac{1 - \epsilon}{\epsilon} \right)^{N_0 - N_1} > 1,$$

e portanto, $\ddot{X}_t(v_0^T, l_0^T) = 0$, para todo $t \in [\tau_{i-1}^t; \tau_i^t - 1)$. Mas,

$$\begin{aligned} \sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} \mathbf{1}_{\{v_s=1\}} &= \sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} (1 - \mathbf{1}_{\{v_s=0\}}) = \\ &= \tau_i^t - \tau_{i-1}^t - \sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} \mathbf{1}_{\{v_s=0\}} \\ &= l_i(t) - \sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} \mathbf{1}_{\{v_s=0\}}. \end{aligned}$$

Logo, $N_0 - N_1 > 0$ implica que

$$\sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} \mathbf{1}_{\{v_s=0\}} > \frac{l_i(t)}{2}.$$

De modo análogo, temos que se $N_0 < N_1$, então $\ddot{X}_t(v_0^T, l_0^T) = 1$, para todo $t \in [\tau_{i-1}^t; \tau_i^t - 1)$.

E se $N_0 = N_1$, então $\ddot{X}_t(v_0^T, l_0^T)$ é escolhido ao acaso. ■

3.1 Número de operações para encontrar

$$\ddot{X}_0^T(v_0^T, l_0^T)$$

Escolher a seqüência x_0^T de estados ocultos que maximiza a

$$P_\theta(X_0^T = x_0^T | V_0^T = v_0^T, L_0^T = l_0^T),$$

para $v_0^T \in B^T$ e $l_0^T \in \{0, 1\}^T$ fixados, esbarra no problema do número de candidatos que é $|A|^T$. Logo o número de operações para escolher o estimador de máxima verossimilhança cresce exponencialmente com o tamanho da amostra. No teorema 3.4, mostramos que o estimador proposto nesta seção, \ddot{X}_0^T , utiliza um número de operações que cresce polinomialmente com o tamanho da amostra.

Vamos calcular o número de operações (η_T) para encontrar $\ddot{X}_0^T(v_0^T, l_0^T)$.

Temos (J_T+1) blocos e o número de cálculos dentro de cada bloco depende do comprimento do bloco, ou seja, l_i .

Vale ressaltar que l_i é uma variável aleatória com distribuição geométrica

com parâmetro λ_0 .

Logo, para encontrar $\ddot{X}_{\tau_j}^{\tau_{j+1}-1}$ temos que calcular

$$P_\theta \left(X_{\tau_j}^{\tau_{j+1}-1} = x_{\tau_j}^{\tau_{j+1}-1}, V_{\tau_j}^{\tau_{j+1}-1} = v_{\tau_j}^{\tau_{j+1}-1}, L_{\tau_j}^{\tau_{j+1}-1} = l_{\tau_j}^{\tau_{j+1}-1} \right), \quad (3.2)$$

$N^{l_{i+1}}$ vezes, onde $N = |A|$. Mas, para calcular (3.2) é preciso realizar $2(l_{i+1} + 1)$ cálculos, logo para encontrar $\ddot{x}_{\tau_j}^{\tau_{j+1}-1}$ é necessário $2(l_{i+1} + 1)N^{l_{i+1}}$ cálculos. Assim podemos concluir que

$$\eta_T = \sum_{i=0}^{J_T} 2(l_{i+1} + 1)N^{l_{i+1}}. \quad (3.3)$$

Teorema 3.4 *Dados v_0^T, l_0^T e θ existe $1 - \frac{2 \ln N}{\ln \lambda_1} < \delta < +\infty$ tal que*

$$\lim_{T \rightarrow +\infty} P \left(\frac{\eta_T}{T^\delta} > 1 \right) = 0, \quad (3.4)$$

onde $|A| = N$ e $P(L_t = 1) = \lambda_1$.

Prova: Por (3.3) temos que

$$\begin{aligned} P \left(\frac{\eta_T}{T^\delta} > 1 \right) &= P \left(\frac{\sum_{i=1}^{J_T+1} 2(l_i + 1) N^{l_i}}{T^\delta} > 1 \right) \\ &\leq 1 - P \left(\frac{\sum_{i=1}^T 2(l_i + 1) N^{l_i}}{T^\delta} \leq 1 \right) \\ &\leq 1 - P \left(2(l_i + 1) N^{l_i} \leq T^{\delta-1}, i = 1, \dots, T \right) \\ &= 1 - \left[P \left(2(l_i + 1) N^{l_i} \leq T^{\delta-1} \right) \right]^T \\ &\leq 1 - \left[P \left(l_i \leq \frac{\ln T^{\delta-1} - \ln 2}{2 \ln N} \right) \right]^T, \end{aligned}$$

considerando $k_T = \lfloor \frac{\ln T^{\delta-1} - \ln 2}{2 \ln N} \rfloor$ temos que

$$\begin{aligned}
 P\left(\frac{\eta_T}{T^\delta} > 1\right) &\leq 1 - \left[\sum_{j=0}^{k_T} \lambda_0 (1 - \lambda_0)^j \right]^T \\
 &= 1 - \lambda_0^T \left[\sum_{j=0}^{k_T} (1 - \lambda_0)^j \right]^T \\
 &= 1 - \lambda_0^T \left(\frac{1}{\lambda_0} - \frac{(1 - \lambda_0)^{k_T}}{\lambda_0} \right)^T \\
 &= 1 - \left(1 - \lambda_1^{k_T} \right)^T \\
 &\leq 1 - 1 + T \lambda_1^{k_T} \\
 &= T \lambda_1^{k_T}.
 \end{aligned}$$

Logo, queremos $\delta > 1$ tal que

$$\lim_{T \rightarrow +\infty} T \lambda_1^{k_T} = 0,$$

mas tomando $\lambda_1 = e^{-c}$ temos que

$$\begin{aligned}
 \lim_{T \rightarrow +\infty} T \lambda_1^{k_T} &= \lim_{T \rightarrow +\infty} e^{\ln T} e^{-ck_T} \\
 &= \lim_{T \rightarrow +\infty} e^{\ln T - ck_T},
 \end{aligned}$$

e portanto, precisamos que

$$\begin{aligned} \ln T &< ck_T \\ \ln T &< c \frac{\ln T^{\delta-1} - \ln 2}{2 \ln N} \\ \frac{2 \ln N}{c} \ln T + \ln 2 &< \ln T^{\delta-1} < \ln T^{\delta-1} + \ln 2 \\ \frac{2 \ln N}{c} &< \delta - 1. \end{aligned}$$

Ou seja, se

$$\delta > 1 - \frac{2 \ln N}{\ln \lambda_1}, \quad (3.5)$$

então

$$\lim_{T \rightarrow +\infty} P\left(\frac{\eta_T}{T^\delta} > 1\right) = 0.$$

Note que $\delta > 1$, pois $\frac{2 \ln N}{\ln \lambda_1} < 0$. ■

No corolário 3.5 vamos mostrar que para o caso $|A| = 2$, o teorema vale para $\delta = 1$, ou seja, o número de operações para encontrar o estimador cresce linearmente com o tamanho da amostra.

Corolário 3.5 *Dados v_0^T, l_0^T e θ temos que*

$$\lim_{T \rightarrow +\infty} P\left(\frac{\eta_T}{T} > 1\right) = 0, \quad (3.6)$$

com $|A| = 2$.

Prova: Quando $|A| = 2$, de acordo com o corolário 2.11, temos que testar apenas 2 seqüências ocultas para encontrar a de máxima verossimilhança.

Portanto, neste caso

$$\eta_T = \sum_{i=0}^{J_T} 2(l_{i+1} + 4). \quad (3.7)$$

E portanto,

$$\begin{aligned} P\left(\frac{\eta_T}{T} > 1\right) &= P\left(\frac{\sum_{i=1}^{J_T+1} 2(l_i + 4)}{T} > 1\right) \\ &\leq 1 - P\left(\frac{\sum_{i=1}^T 2(l_i + 4)}{T} \leq 1\right) \\ &\leq 1 - P(2(l_i + 4) \leq T, i = 1, \dots, T) \\ &= 1 - [P(2(l_i + 4) \leq T)]^T \\ &\leq 1 - \left[P\left(l_i \leq \frac{T}{2} - 4\right)\right]^T. \end{aligned}$$

Considerando agora $k_T = \lfloor \frac{T}{2} - 4 \rfloor$ e usando os mesmos argumentos da prova do teorema 3.4 temos que

$$\begin{aligned} P\left(\frac{\eta_T}{T^\delta} > 1\right) &\leq T\lambda_1^{k_T} \\ &= T\lambda_1^{\frac{T}{2}-4}. \end{aligned}$$

Mas,

$$\lim_{T \rightarrow +\infty} T\lambda_1^{k_T} = 0.$$

■

3.2 Erro médio assintótico de $\ddot{X}_t(v_0^T, l_0^T)$

Vamos começar esta seção demonstrando um resultado sobre a esperança do erro do estimador $\ddot{X}_t(v_0^T, l_0^T)$ num ponto t fixo, dados os instantes de regeneração e a seqüência visível.

O lema a seguir nos mostra que a probabilidade de erro no instante t depende do bloco ao qual ele pertence, isto é, a probabilidade de erro é constante para todos os instantes pertencentes ao bloco.

Lema 3.6 *Dados v_0^T e l_0^T , para todo $t \in [\tau_{i-1}^t; \tau_i^t)$, $i = 1, \dots, J_T$, sendo $\tau_{J_T} = T$*

$$E\left(\mathbf{1}_{\{\ddot{X}_t(v_0^T, l_0^T) \neq X_t\}} \mid V_0^T = v_0^T, L_0^T = l_0^T\right) \quad (3.8)$$

é dada por

$$\sum_{l=\frac{l_i(t)}{2}+1}^{l_i(t)} \binom{l_i(t)}{l} \epsilon^l (1-\epsilon)^{l_i(t)-l} + \frac{1}{2} \binom{l_i(t)}{l_i(t)/2} [\epsilon(1-\epsilon)]^{\frac{l_i(t)}{2}} \mathbf{1}_{\{l_i(t) \text{ ser par}\}},$$

com $l_i(t) = \tau_i^t - \tau_{i-1}^t$.

Prova: Dado l_0^T , sabemos que existe $i \in \{1, \dots, J_T\}$ tal que $\tau_{i-1}^t \leq t < \tau_i^t$.

Pelo lema 3.3, temos que para todo $t \in [\tau_{i-1}^t; \tau_i^t)$

$$\begin{aligned} \left[\ddot{X}_t(v_0^T, l_0^T) \neq X_t\right] &= \left[X_t = x, N_{1-x} > \frac{l_i(t)}{2}\right] \cup \\ &\quad \left[l_i(t) \text{ ser par}, X_t = x, N_x = \frac{l_i(t)}{2}, U > 1/2\right], \end{aligned}$$

ou seja,

$$\begin{aligned} \left[\ddot{X}_t(v_0^T, l_0^T) \neq X_t \right] &= \left[\sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} \mathbf{1}_{\{X_s \neq v_s\}} > \frac{l_i(t)}{2} \right] \cup \\ &\left[l_i(t) \text{ ser par, } \sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} \mathbf{1}_{\{X_s \neq v_s\}} = \frac{l_i(t)}{2}, U > 1/2 \right], \end{aligned}$$

que, pela definição 2.7, é igual a

$$\left[\sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} (1 - \delta_s) > \frac{l_i(t)}{2} \right] \cup \left[l_i(t) \text{ ser par, } \sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} (1 - \delta_s) = \frac{l_i(t)}{2}, U > 1/2 \right].$$

Neste caso, (3.8) é dado por

$$P_\theta \left(\sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} (1 - \delta_s) > \frac{l_i(t)}{2} \right) + \frac{1}{2} P_\theta \left(\sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} (1 - \delta_s) = \frac{l_i(t)}{2} \right) \mathbf{1}_{\{l_i(t) \text{ ser par}\}}.$$

Mas como, por (2.10), δ_s é uma Bernoulli de parâmetro ϵ , então

$$\left[\sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} (1 - \delta_s) \right] \sim \beta(l_i(t), \epsilon).$$

Portanto,

$$P_\theta \left(\sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} (1 - \delta_s) > \frac{l_i(t)}{2} \right) = \sum_{l=\frac{l_i(t)}{2}+1}^{l_i(t)} \binom{l_i(t)}{l} \epsilon^l (1 - \epsilon)^{l_i(t)-l}$$

e,

$$P_\theta \left(\sum_{s=\tau_{i-1}^t}^{\tau_i^t-1} (1 - \delta_s) = \frac{l_i(t)}{2}, U > 1/2 \right) = \frac{1}{2} \binom{l_i(t)}{l_i(t)/2} [\epsilon(1 - \epsilon)]^{\frac{l_i(t)}{2}},$$

o que conclui a demonstração. \blacksquare

O segundo resultado desta seção diz respeito ao número médio de erros do estimador $\ddot{X}_t(v_0^T, l_0^T)$.

Teorema 3.7 *Dados v_0^T, l_0^T , para todo $\gamma > 0$ temos que*

$$P_\theta \left(\left| \frac{1}{T+1} \sum_{t=0}^T \mathbf{1}_{\{\ddot{X}_t(v_0^T, l_0^T) \neq X_t\}} - p \right| > \gamma \mid V_0^T = v_0^T, L_0^T = l_0^T \right) \leq 2 \exp^{-2\gamma^2(T+1)c(l_0^T)},$$

onde

- $p = \frac{\sum_{i=1}^{J_T} l_i p_i}{T+1}$;
- $p_i = E \left(\mathbf{1}_{\{\ddot{X}_t(v_0^T, l_0^T) \neq X_t\}} \mid V_0^T = v_0^T, L_0^T = l_0^T \right)$, $\tau_{i-1}^t \leq t < \tau_i^t$;
- $c(l_0^T) = \left[\frac{J_T}{T+1} \frac{1}{J_T} \sum_{i=1}^{J_T} l_i^2(t) \right]^{-1}$.

Prova: Definindo, para todo $i \in \{1, \dots, J_T\}$

$$Y_i = \mathbf{1}_{\{\ddot{X}_t(v_0^T, l_0^T) \neq X_t, \tau_{i-1}^t \leq t < \tau_i^t\}},$$

temos, pela independência dos blocos, que Y_i são independentes. Podemos

então reescrever $\frac{\sum_{t=0}^T \mathbf{1}_{\{\ddot{X}_t(v_0^T, l_0^T) \neq X_t\}}}{T+1} - p$ da seguinte maneira

$$\begin{aligned} \frac{\sum_{t=0}^T \mathbf{1}_{\{\ddot{X}_t(v_0^T, l_0^T) \neq X_t\}}}{T+1} - p &= \frac{\sum_{i=1}^{J_T} l_i \mathbf{1}_{\{\ddot{X}_t(v_0^T, l_0^T) \neq X_t, \tau_{i-1}^t \leq t < \tau_i^t\}}}{T+1} - \frac{\sum_{i=1}^{J_T} l_i p_i}{T+1} \\ &= \frac{\sum_{i=1}^{J_T} l_i Y_i}{T+1} - \frac{\sum_{i=1}^{J_T} l_i p_i}{T+1} \\ &= \frac{1}{T+1} \sum_{i=1}^{J_T} l_i (Y_i - p_i). \end{aligned}$$

Logo,

$$\begin{aligned} &P_\theta \left(\left| \frac{\sum_{t=0}^T \mathbf{1}_{\{\ddot{X}_t(v_0^T, l_0^T) \neq X_t\}}}{T+1} - p \right| > \gamma \mid V_0^T = v_0^T, L_0^T = l_0^T \right) \\ &= P_\theta \left(\left| \frac{\sum_{i=1}^{J_T} l_i (Y_i - p_i)}{T+1} \right| > \gamma \mid V_0^T = v_0^T, L_0^T = l_0^T \right). \end{aligned}$$

Fazendo $W_i = l_i Y_i$, temos que $0 \leq W_i \leq l_i$ e

$$E(W_i \mid V_0^T = v_0^T, L_0^T = l_0^T) = l_i p_i,$$

e portanto, pela desigualdade de Hoeffding (ver apêndice A), temos que

$$\begin{aligned} &P_\theta \left(\left| \frac{\sum_{t=0}^T \mathbf{1}_{\{\ddot{X}_t(v_0^T, l_0^T) \neq X_t\}}}{T+1} - p \right| > \gamma \mid V_0^T = v_0^T, L_0^T = l_0^T \right) \\ &\leq 2 \exp^{-2\gamma^2} \frac{(T+1)^2}{\sum_{i=1}^{J_T} l_i^2(t)} \\ &= 2 \exp^{-2\gamma^2(T+1)c(l_0^T)}, \end{aligned}$$

como queríamos demonstrar. ■

Capítulo 4

Estimação de X_0^T por blocos de consenso

O problema do estimador $\ddot{X}_0^T(v_0^T, l_0^T)$ é que não conhecemos a seqüência $(L_t)_{t \geq 0}$ que originou a seqüência oculta.

O estimador proposto vai utilizar uma seqüência $(\tilde{L}_t)_{t \in \mathcal{Z}}$ que é escolhida de maneira independente da cadeia oculta, com a seguinte lei:

$$P(\tilde{L}_t = 0) = \tilde{\lambda}_0.$$

A seqüência estimada será escolhida em blocos definidos por $(\tilde{L}_t)_{t \in \mathcal{Z}}$.

Este estimador também é escolhido por blocos, como no caso do estimador $\ddot{X}_0^T(v_0^T, l_0^T)$, logo o número de operações para encontrar o estimador também cresce linearmente com o tamanho da amostra, fato que explicaremos na seção 4.2.

Vamos começar definindo algumas variáveis aleatórias que utilizaremos

posteriormente, e enunciando lemas sobre as distribuições dessas variáveis.

Definição 4.1 *Vamos definir $\tilde{\tau}_0^t$, $\tilde{\tau}_1^t$ e $\tilde{l}(t)$ para todo $t \in \mathcal{Z}$.*

- *Seja $\tilde{\tau}_0^t \leq t$ definido por*

$$\tilde{\tau}_0^t = \max\{s : \tilde{L}_s = 0, s \leq t\}.$$

- *Seja $\tilde{\tau}_1^t \geq t + 1$ definido por*

$$\tilde{\tau}_1^t = \min\{s : \tilde{L}_s = 0, s > t\}.$$

- *Seja $\tilde{l}(t) \in \{1, 2, \dots\}$ definido por*

$$\tilde{l}(t) = \tilde{\tau}_1^t - \tilde{\tau}_0^t.$$

A seguir descrevemos as distribuições de $\tilde{\tau}_0^t$, $\tilde{\tau}_1^t$ e $\tilde{l}(t)$ que serão obtidas a partir dos lemas 4.2, 4.3 e 4.4, respectivamente.

Lema 4.2 *Seja $\tilde{\tau}_0^t$ como na definição 4.1, então*

$$P(\tilde{\tau}_0^t = s) = \tilde{\lambda}_0 \tilde{\lambda}_1^{t-s}, \text{ se } s \leq t.$$

Prova: O resultado é decorrência direta da independência entre os \tilde{L}_t , pois $\forall s \leq t$

$$\begin{aligned} P(\tilde{\tau}_0^t = s) &= P\left(\tilde{L}_s = 0, \tilde{L}_r = 1, r = s + 1, s + 2, \dots, t\right) \\ &= P\left(\tilde{L}_s = 0\right) \left[P\left(\tilde{L}_r = 1\right)\right]^{t-s}. \end{aligned}$$

O que conclui a demonstração. ■

Lema 4.3 *Seja $\tilde{\tau}_1^t$ como na definição 4.1, então*

$$P(\tilde{\tau}_1^t = s) = \tilde{\lambda}_0 \tilde{\lambda}_1^{s-t-1}, \text{ se } s \geq t + 1.$$

Prova: Análoga à prova do lema 4.2. ■

Lema 4.4 *Seja $\tilde{l}(t)$ como na definição 4.1, então*

$$P(\tilde{l}(t) = l) = l \tilde{\lambda}_0^2 \tilde{\lambda}_1^{l-1}, \text{ se } l \geq 1.$$

Prova: Pela independência entre os \tilde{L}_t , temos que $\tilde{\tau}_0^t$ e $\tilde{\tau}_1^t$ também são independentes, e portanto pelos lemas 4.2 e 4.3

$$\begin{aligned} P(\tilde{l}(t) = l) &= \sum_{k=t-l+1}^t P(\tilde{\tau}_0^t = k) P(\tilde{\tau}_1^t = l+k) \\ &= \sum_{k=t-l+1}^t \tilde{\lambda}_0 \tilde{\lambda}_1^{t-k} \tilde{\lambda}_0 \tilde{\lambda}_1^{l+k-t-1} \\ &= l \tilde{\lambda}_0^2 \tilde{\lambda}_1^{l-1}. \end{aligned}$$

Agora podemos definir $\hat{X}_t(v_0^T, \tilde{l}_0^T)$. ■

Definição 4.5 *De acordo com a definição 4.1 definimos*

$$\hat{X}_t(v_0^T, \tilde{l}_0^T) = \begin{cases} 0 & \text{se } \sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} \mathbf{1}_{\{v_s=0\}} \geq \frac{\tilde{l}(t)}{2} \\ 1 & \text{se } \sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} \mathbf{1}_{\{v_s=1\}} > \frac{\tilde{l}(t)}{2} \\ \mathbf{1}_{\{U>1/2\}} & \text{se } \sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} \mathbf{1}_{\{v_s=0\}} = \frac{\tilde{l}(t)}{2}, \end{cases}$$

com $U \sim \text{Uniforme}(0; 1)$.

4.1 Algoritmo para encontrar \hat{X}_t

Para obtermos $\hat{X}_t(v_0^T, \tilde{l}_0^T)$ utilizamos o seguinte algoritmo:

1. sorteamos uma seqüência \tilde{L}_0^T , com $P(\tilde{L}_t = 0) = \tilde{\lambda}_0$;
2. para cada bloco definido pela seqüência $(\tilde{L}_t)_{t \geq 0}$ escolhe-se a seqüência oculta de acordo com a definição 4.5.

4.2 Número de operações para encontrar \hat{X}_0^T

Para encontrar $\hat{X}_0^T(v_0^T, \tilde{l}_0^T)$ temos que calcular

$$\sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} \mathbf{1}_{\{v_s=0\}} \quad (4.1)$$

para cada bloco, isto é, $J_T + 1$ vezes. Mas, para calcular (4.1) é preciso realizar $\tilde{l}(t)$ somas, com isso podemos concluir que o número de operações para encontrar $\hat{X}_0^T(v_0^T, \tilde{l}_0^T)$ é a soma dos comprimentos dos $J_T + 1$ blocos que é exatamente o comprimento da seqüência.

4.3 Erro médio assintótico de \hat{X}_t

Começaremos esta seção apresentado dois lemas que serão utilizados na demonstração dos resultados desta seção.

Lema 4.6 *Sejam $u \leq 1$ e/ou $b \leq 1$, então*

$$\sum_{l \geq 1} lu^l b^{l-1} = \frac{u}{(1-ub)^2}.$$

Prova: Como $u \leq 1$ e/ou $b \leq 1$, sabemos que $\sum_{l \geq 1} (ub)^l$ converge e portanto

$$\frac{d}{db} \sum_{l \geq 1} (ub)^l = \sum_{l \geq 1} \frac{d}{db} (ub)^l = \sum_{l \geq 1} lu^l b^{l-1}. \quad (4.2)$$

Deste modo,

$$\begin{aligned} \sum_{l \geq 1} lu^l b^{l-1} &= \frac{d}{db} \sum_{l \geq 1} (ub)^l = \frac{d}{db} \frac{ub}{1-ub} \\ &= \frac{u}{(1-ub)^2}. \end{aligned}$$

■

A seguir enunciaremos o teorema 4.7 que apresenta um resultado sobre a probabilidade do estimador $\hat{X}_t(v_0^T, \tilde{l}_0^T)$ não escolher o verdadeiro valor oculto.

Teorema 4.7 *Seja $\hat{X}_0^T (v_0^T, \tilde{l}_0^T)$ como definido anteriormente, então*

$$\begin{aligned} \tilde{E} \left(E \left(\mathbf{1}_{\{\hat{X}_t(v_0^T, \tilde{l}_0^T) \neq X_t\}} \right) \right) &\leq 1 + \tilde{\lambda}_0^2 \left\{ - \frac{1}{\left[1 - \tilde{\lambda}_1(1-q) \right]^2} + \right. \\ &\quad + \frac{e^{2\epsilon-1}}{\left[1 - \tilde{\lambda}_1(1-q)e^{2\epsilon-1} \right]^2} + \\ &\quad \left. + \frac{2[\epsilon(1-\epsilon)]^{1/2}}{\left\{ 1 - 2\tilde{\lambda}_1(1-q)[\epsilon(1-\epsilon)]^{1/2} \right\}^2} \right\} \end{aligned}$$

para quase toda escolha de (v_0^T, \tilde{l}_0^T) .

Prova: Para simplificar a notação vamos substituir $\hat{X}_t (v_0^T, \tilde{l}_0^T)$ por \hat{X}_t . Vamos começar calculando $E \left(\mathbf{1}_{\{\hat{X}_t \neq X_t\}} \right)$. Mas, usando

$$\left\{ \sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-2} \mathbf{1}_{\{X_s \neq X_{s+1}\}} = k, k \geq 0 \right\}$$

como partição, temos que

$$E \left(\mathbf{1}_{\{\hat{X}_t \neq X_t\}} \right) = \sum_{k=0}^{\tilde{l}(t)-1} P_\theta \left(\hat{X}_t \neq X_t, \sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-2} \mathbf{1}_{\{X_s \neq X_{s+1}\}} = k \right). \quad (4.3)$$

Vamos calcular o primeiro termo de (4.3), que corresponde à situação em que todos os estados da seqüência visível entre $\tilde{\tau}_0^t$ e $\tilde{\tau}_1^t$ são iguais entre si. Assim, para $k = 0$, aplicando a definição 4.5, temos

$$\begin{aligned}
\sum_{x=0,1} P_\theta \left(\hat{X}_t \neq X_t, X_{\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} = x \right) &= \sum_{x=0,1} P_\theta \left(\hat{X}_t \neq X_t \middle| X_{\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} = x \right) * \\
&* P_\theta \left(X_{\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} = x \right) = \\
&= \sum_{x=0,1} \left[P_\theta \left(\sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} \mathbf{1}_{\{V_s=1-x\}} > \frac{\tilde{l}(t)}{2} \middle| X_{\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} = x \right) + \right. \\
&+ \left. \frac{1}{2} P_\theta \left(\sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} \mathbf{1}_{\{V_s=1-x\}} = \frac{\tilde{l}(t)}{2} \middle| X_{\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} = x \right) \right] * \\
&* P_\theta \left(X_{\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} = x \right).
\end{aligned}$$

Mas, pela desigualdade de Hoeffding (ver apêndice A),

$$P_\theta \left(\sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} \mathbf{1}_{\{V_s=1-x\}} > \frac{\tilde{l}(t)}{2} \middle| X_{\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} = x \right) \leq e^{-2\tilde{l}(t)(1/2-\epsilon)}$$

e, pelo lema 2.13 e usando a aproximação de Stirling, temos que

$$\begin{aligned}
\frac{1}{2} P_\theta \left(\sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} \mathbf{1}_{\{V_s=1-x\}} = \frac{\tilde{l}(t)}{2} \middle| X_{\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} = x \right) &= \frac{1}{2} \binom{\tilde{l}(t)}{\tilde{l}(t)/2} [\epsilon(1-\epsilon)]^{\tilde{l}(t)/2} \leq \\
&\leq 2^{\tilde{l}(t)} \tilde{l}(t)^{-1/2} [\epsilon(1-\epsilon)]^{\tilde{l}(t)/2} = \\
&= \tilde{l}(t)^{-1/2} \{2[\epsilon(1-\epsilon)]^{1/2}\}^{\tilde{l}(t)}.
\end{aligned}$$

Logo, como $P_\theta \left(X_{\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-1} = x \right) = \frac{(1-q)^{\tilde{l}(t)-1}}{2}$, temos que

$$\begin{aligned}
& P_\theta \left(\hat{X}_t \neq X_t, \sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-2} \mathbf{1}_{\{X_s \neq X_{s+1}\}} = 0 \right) \\
& \leq (1-q)^{\tilde{l}(t)-1} \left(e^{-\tilde{l}(t)(1-2\epsilon)} + \tilde{l}(t)^{-1/2} \{2[\epsilon(1-\epsilon)]^{1/2}\}^{\tilde{l}(t)} \right). \quad (4.4)
\end{aligned}$$

Para $k \geq 1$, isto é, quando 1 ou mais fronteiras de blocos da seqüência oculta está contida no bloco escolhido, usando o lema 2.13, temos

$$\begin{aligned}
P_\theta \left(\hat{X}_t \neq X_t, \sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-2} \mathbf{1}_{\{X_s \neq X_{s+1}\}} \geq 1 \right) & \leq P_\theta \left(\sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-2} \mathbf{1}_{\{X_s \neq X_{s+1}\}} \geq 1 \right) \quad (4.5) \\
& = 1 - P_\theta \left(\sum_{s=\tilde{\tau}_0^t}^{\tilde{\tau}_1^t-2} \mathbf{1}_{\{X_s \neq X_{s+1}\}} = 0 \right) \\
& = 1 - (1-q)^{\tilde{l}(t)-1}.
\end{aligned}$$

Deste modo, por (4.4) e (4.5), temos que

$$E \left(\mathbf{1}_{\{\hat{X}_t \neq X_t\}} \right) \leq 1 + (1-q)^{-1} \left\{ - (1-q)^{\tilde{l}(t)} + [(1-q)e^{2\epsilon-1}]^{\tilde{l}(t)} \right. \quad (4.6)$$

$$\left. + \tilde{l}(t)^{-1/2} \{2(1-q)[\epsilon(1-\epsilon)]^{1/2}\}^{\tilde{l}(t)} \right\}. \quad (4.7)$$

Portanto,

$$\begin{aligned}
\tilde{E} \left(E \left(\mathbf{1}_{\{\hat{X}_t \neq X_t\}} \right) \right) & \leq 1 + (1-q)^{-1} \tilde{E} \left(- (1-q)^{\tilde{l}(t)} + [(1-q)e^{2\epsilon-1}]^{\tilde{l}(t)} \right. \\
& \left. + \tilde{l}(t)^{-1/2} \{2(1-q)[\epsilon(1-\epsilon)]^{1/2}\}^{\tilde{l}(t)} \right).
\end{aligned}$$

Vamos agora calcular a esperança de cada um dos termos dessa soma. Usando os lemas 4.4 e 4.6, temos que

$$\begin{aligned} \tilde{E} \left((1-q)^{\tilde{l}(t)} \right) &= \sum_{l \geq 1} (1-q)^l P(\tilde{l}(t) = l) = \sum_{l \geq 1} (1-q)^l l \tilde{\lambda}_0^2 \tilde{\lambda}_1^{l-1} \\ &= \frac{\tilde{\lambda}_0^2 (1-q)}{\left[1 - \tilde{\lambda}_1 (1-q) \right]^2}, \end{aligned} \quad (4.8)$$

analogamente,

$$\tilde{E} \left(\left[(1-q)e^{2\epsilon-1} \right]^{\tilde{l}(t)} \right) = \frac{\tilde{\lambda}_0^2 (1-q) e^{2\epsilon-1}}{\left[1 - \tilde{\lambda}_1 (1-q) e^{2\epsilon-1} \right]^2} \quad (4.9)$$

e

$$\tilde{E} \left(\tilde{l}(t)^{-1/2} \left\{ 2(1-q)[\epsilon(1-\epsilon)]^{1/2} \right\}^{\tilde{l}(t)} \right) \leq \frac{\tilde{\lambda}_0^2 \left\{ 2(1-q)[\epsilon(1-\epsilon)]^{1/2} \right\}}{\left[1 - \tilde{\lambda}_1 \left\{ 2(1-q)[\epsilon(1-\epsilon)]^{1/2} \right\} \right]^2} \quad (4.10)$$

Logo, por (4.8), (4.9) e (4.10)

$$\begin{aligned} \tilde{E} \left(E \left(\mathbf{1}_{\{\hat{X}_t \neq X_t\}} \right) \right) &\leq 1 + (1-q)^{-1} \left\{ - \frac{\tilde{\lambda}_0^2 (1-q)}{\left[1 - \tilde{\lambda}_1 (1-q) \right]^2} + \right. \\ &\quad + \frac{\tilde{\lambda}_0^2 (1-q) e^{2\epsilon-1}}{\left[1 - \tilde{\lambda}_1 (1-q) e^{2\epsilon-1} \right]^2} + \\ &\quad \left. + \frac{\tilde{\lambda}_0^2 \left\{ 2(1-q)[\epsilon(1-\epsilon)]^{1/2} \right\}}{\left[1 - \tilde{\lambda}_1 \left\{ 2(1-q)[\epsilon(1-\epsilon)]^{1/2} \right\} \right]^2} \right\} = \end{aligned}$$

$$\begin{aligned}
&= 1 + \tilde{\lambda}_0^2 \left\{ - \frac{1}{\left[1 - \tilde{\lambda}_1(1 - q)\right]^2} + \right. \\
&\quad + \frac{e^{2\epsilon-1}}{\left[1 - \tilde{\lambda}_1(1 - q)e^{2\epsilon-1}\right]^2} + \\
&\quad \left. + \frac{2[\epsilon(1 - \epsilon)]^{1/2}}{\left\{1 - 2\tilde{\lambda}_1(1 - q)[\epsilon(1 - \epsilon)]^{1/2}\right\}^2} \right\}.
\end{aligned}$$

O que conclui a demonstração. ■

4.4 Estimação da seqüência oculta usando a moda de \hat{X}_t

Para diminuir o erro médio vamos propor um estimador baseado na moda de $\hat{X}_t(v_0^T, \tilde{l}_0^T)$.

Definição 4.8 *De acordo com a definição 4.5, podemos definir o estimador da moda por:*

$$\hat{X}_t^{(K)}(v_0^T) = \text{moda} \left\{ \hat{X}_t^{(k)} : k = 1, 2, \dots, K \right\},$$

com $\hat{X}_t^{(k)}$ é $\hat{X}_t(v_0^T, \tilde{l}_0^T)$ calculado pela k -ésima vez.

Podemos notar, pelo lema 4.7 que

$$\tilde{E} \left(E \left(\mathbf{1}_{\{\hat{X}_t \neq X_t\}} \right) \right)$$

é uma função de $\tilde{\lambda}_0$, q e ϵ . Na seção 6.1 apresentamos simulações que mostram a influência desses parâmetros nos estimadores $\hat{X}_t(v_0^T, \tilde{t}_0^T)$ e $\hat{X}_t^{(K)}(v_0^T)$.

Capítulo 5

Estimação de X_0^T usando método Monte Carlo markoviano

Neste capítulo vamos propor um estimador baseado no método Monte Carlo markoviano com algoritmo de Metrópolis. A definição a seguir formaliza tal estimador.

Definição 5.1 *Dado $v_0^T \in B^T$, então*

$$\bar{X}_0^T(v_0^T) = \underset{(x_0^T, l_0^T) \in A^T \times \{0,1\}^T}{\operatorname{argmax}} P_\theta(X_0^T = x_0^T, L_0^T = l_0^T | V_0^T = v_0^T). \quad (5.1)$$

Nesta seção também vamos trabalhar com a restrição $A = B = \{0, 1\}$, mas não com $q(0|0) = q(1|1)$ e $m_0(0) = m_1(1)$. Aqui apenas impomos que $q(x|x) > q(x|y)$ e $m_x(x) = 1 - \delta_x$, $x \in A$.

Novamente, temos que o argumento do máximo de (5.1) é o mesmo de $P_\theta (V_0^T = v_0^T, X_0^T = x_0^T, L_0^T = l_0^T)$, que é igual a

$$\lambda_0^{J_T} (1 - \lambda_0)^{T - J_T} \prod_{i=0}^{J_T} p^0(x_{t_i}) (1 - \delta_{x_{t_i}})^{N_i(v_0^T)} \delta_{x_{t_i}}^{\Delta_i(l_0^T) - N_i(v_0^T)}, \quad (5.2)$$

se $x_s = x_{t_i}$, $s = t_i + 1, \dots, t_{i+1} - 1$ e onde

$$N_i(v_0^T) = \sum_{s=t_i}^{t_{i+1}-1} \mathbf{1}\{v_s = x_{t_i}\}, \quad \Delta_i(l_0^T) = t_{i+1} - t_i$$

e $\tau_i = t_i$.

Definição 5.2 *Seja $v = v_0^T$, definimos a função*

$$F_v : A^T \times \{0, 1\}^T \rightarrow \mathbf{R}^+$$

por

$$F_v(x_0^T, l_0^T) = \begin{cases} P_\theta(V_0^T = v_0^T, X_0^T = x_0^T, L_0^T = l_0^T), & \text{se } \begin{cases} x_s = x_{t_i}, s = t_i + 1, \dots, t_{i+1} - 1, \\ i = 0, \dots, J_T \end{cases} \\ 0, & \text{caso contrário} \end{cases}$$

Definição 5.3 *Seja $v = v_0^T$ e $\beta > 0$, então $\mu_{v,\beta}$, medida de probabilidade, é definida por:*

$$\mu_{v,\beta}(x_0^T, l_0^T) = \frac{e^{\beta \log F_v(x_0^T, l_0^T)}}{Z_{v,\beta}}, \quad (5.3)$$

onde

$$Z_{v,\beta} = \sum_{(z_0^T, r_0^T) \in A^T \times \{0, 1\}^T} e^{\beta \log F_v(z_0^T, r_0^T)}.$$

Podemos observar que o par (x_0^T, l_0^T) que maximiza $\mu_{v,\beta}(x_0^T, l_0^T)$ é o mesmo que maximiza $F_v(x_0^T, l_0^T)$. Mas, como é difícil calcular $Z_{v,\beta}$ usamos uma cadeia de Markov que seja reversível com relação a $\mu_{v,\beta}$. Vamos a seguir definir tal cadeia.

Definição 5.4 *Seja $(\xi(k))_{k \geq 1}$ uma cadeia de Markov, com $\xi(k) = (x_0^T(k), l_0^T(k))$ e a seguinte matriz de transição:*

$$P(\xi|\eta) = \left\{ \begin{array}{ll} \frac{1}{T} [\lambda_0^\beta J_T(r_0^T) + \lambda_1^\beta (T - J_T(r_0^T))], & \text{se } l_t = r_t, \forall t \\ \frac{1}{T} \frac{\lambda_1^\beta}{\lambda_0^\beta + \lambda_1^\beta}, & \text{se } \left\{ \begin{array}{l} l_s = 1, r_s = 0, \\ l_t = r_t, \forall t \neq s, \\ x_t = y_t, \forall t \end{array} \right. \\ \frac{1}{T} \frac{\lambda_1^\beta}{\lambda_0^\beta + \lambda_1^\beta} \min \left\{ 1, \frac{1}{p^0(y_s)} \left(\frac{1 - \delta_{x_s}}{\delta_{y_s}} \right)^{\Delta_s - N_s(y)} \left(\frac{\delta_{x_s}}{1 - \delta_{y_s}} \right)^{N_s(y)} \right\}, & \text{se } \left\{ \begin{array}{l} l_s = 1, r_s = 0, \\ l_t = r_t, \forall t \neq s, \\ x_t = 1 - y_t, s \leq t < \tau^* \end{array} \right. \\ \frac{1}{T} \frac{\lambda_0^\beta}{\lambda_0^\beta + \lambda_1^\beta} \min \left\{ 1, p^0(x_s) \left(\frac{1 - \delta_{x_s}}{\delta_{y_s}} \right)^{\Delta_s - N_s(y)} \left(\frac{\delta_{x_s}}{1 - \delta_{y_s}} \right)^{N_s(y)} \right\}, & \text{se } \left\{ \begin{array}{l} l_s = 0, r_s = 1, \\ l_t = r_t, \forall t \neq s \end{array} \right. \\ 0, & \text{caso contrário} \end{array} \right. ,$$

onde $\tau^* = \min\{t > s : l_t(k) = 0\}$, $\Delta_s = \tau^* - s$ e $N_s(y) = \sum_{t=s}^{\tau^*-1} \mathbf{1}\{v_s = y_t\}$.

Para simplificar a notação foi usado que $\xi(k) = (x_0^T(k), l_0^T(k)) = (x_0^T, l_0^T)$ e $\eta = (x_0^T(k-1), l_0^T(k-1)) = (y_0^T, r_0^T)$.

5.1 Algoritmo para encontrar \bar{X}_0^T

O algoritmo que descrevemos a seguir, simula a evolução da cadeia de Markov da definição 5.4.

1. Inicialização

Escolho $l_0^T(0)$ através de Bernoulli independentes todas com parâmetro λ_0 e tomo $x_0^T(0)$ como um vetor de 1's.

2. Indução

No passo $k + 1$

- (a) faço $l_t(k + 1) = l_t(k), \forall t$
- (b) escolho um índice s , ao acaso, uniformemente no conjunto $\{0, 1, \dots, T\}$ e defino τ^*
- (c) gero uma uniforme em $[0,1]$, U_1 , se $U_1 \leq \frac{(\lambda_0)^\beta}{(\lambda_0)^\beta + (\lambda_1)^\beta}$ então $l_s(k + 1) = 0$, senão $l_s(k + 1) = 1$
- (d) se $l_s(k + 1) = l_s(k)$ então $\xi(k + 1) = \xi(k)$, caso contrário
 - i. se $l_s(k + 1) = 0$ então um bloco foi dividido em dois e fazemos o segundo bloco diferente do primeiro, ou seja, $x_t = 1 - x_s(k)$, $s \leq t < \tau^*$.
 - ii. se $l_s(k + 1) = 1$ então dois blocos foram unidos e fazemos o segundo bloco igual ao primeiro, isto é, $x_t = x_{s-1}(k)$, $s \leq t < \tau^*$.

Assim obtemos $\xi = (x_0^T, l_0^T(k + 1))$ onde $x_t = x_t(k)$, se $t \in [s, \tau^* - 1]^c$ e $x_s^{\tau^* - 1}$ foi construído como descrito anteriormente.

- iii. Calculo $R = \left(\frac{\mu_{v,\beta}(\xi)}{\mu_{v,\beta}(\xi(k))} \right)^\beta$
- A. Se $R \geq 1$ então $\xi(k+1) = \xi$
- B. Se $R < 1$ então gero uma uniforme em $[0, 1]$, U_2 , se $U_2 \leq R$ então $\xi(k+1) = \xi$, caso contrário $\xi(k+1) = \xi(k)$

O teorema 5.5 mostra que a cadeia de Markov $(\xi(k))_{k \geq 1}$ é reversível com relação a medida $\mu_{v,\beta}(\cdot)$.

Teorema 5.5 *Sejam $P(\xi|\eta)$ e $(\xi(k))_{k \geq 1}$ conforme a definição 5.4 e $\mu_{v,\beta}(\cdot)$ de acordo com a definição 5.3. Então podemos afirmar que $(\xi(k))_{k \geq 1}$ é reversível com relação a $\mu_{v,\beta}(\cdot)$.*

Prova: É fácil ver que esta matriz de transição é reversível com relação a $\mu_{v,\beta}(\cdot)$, pois decompondo $\mu_{v,\beta}(\cdot)$ e $P(\xi|\eta)$ temos que:

$$\mu_{v,\beta}(\xi) = \frac{1}{Z_{v,\beta}} \mu_L(l_0^T) \mu_V(x_0^T | l_0^T), \quad (5.4)$$

onde

$$\mu_L(l_0^T) = \left(\lambda_0^{J_T(l_0^T)} \lambda_1^{T-J_T(l_0^T)} \right)^\beta,$$

$$\mu_V(x_0^T | l_0^T) = \left(\prod_{i=0}^{J_T} p^0(x_{t_i}) (1 - \delta_{x_{t_i}})^{N_i(v_0^T)} \delta_{x_{t_i}}^{\Delta_i(l_0^T) - N_i(v_0^T)} \right)^\beta$$

e

$$P(\xi|\eta) = p_L(l_0^T | r_0^T) A_V(x_0^T | y_0^T, l_0^T), \quad (5.5)$$

com

$$p_L(l_0^T | r_0^T) = \begin{cases} \frac{1}{T} [\lambda_0^\beta J_T(r_0^T) + \lambda_1^\beta (T - J_T(r_0^T))] & \text{se } l_t = r_t, \forall t \\ \frac{1}{T} \frac{\lambda_i^\beta}{\lambda_0^\beta + \lambda_1^\beta} & \text{se } \begin{cases} l_s = i = 1 - r_s, \\ l_t = r_t, \forall t \neq s \end{cases} \\ 0 & \text{caso contrário} \end{cases}$$

e

$$A_V(x_0^T | y_0^T, l_0^T) = \min \left\{ 1, \frac{\mu_V(x_0^T | l_0^T)}{\mu_V(y_0^T | r_0^T)} \right\}.$$

■

Capítulo 6

Simulações

6.1 Simulações para os estimadores $\hat{X}_t(v_0^T, \tilde{l}_0^T)$ e $\hat{X}_t^{(K)}$

Para avaliar a qualidade dos estimadores $\hat{X}_t(v_0^T, \tilde{l}_0^T)$ e $\hat{X}_t^{(K)}$ realizamos algumas simulações para medir a probabilidade de erro.

A seguir mostramos alguns gráficos que ilustram a probabilidade de erro para alguns valores de q (0,01, 0,1, 0,2, 0,3), ϵ (0,01, 0,1, 0,2, 0,3) e $\tilde{\lambda}_0$ (0,25 - blocos grandes, 0,75 - blocos pequenos).

As simulações foram feitas para uma seqüência de tamanho 1000 e o número de repetições para estimação pela moda variou de 1 a 101, sempre em número ímpares de repetições para evitar empates.

O procedimento foi repetido 5 vezes para cada combinação de q , ϵ e $\tilde{\lambda}_0$.

Através dos gráficos podemos tirar algumas conclusões:

- os gráficos 6.2, 6.3, 6.4, 6.7 e 6.8 mostram que quando q é menor que ϵ ,

no caso das simulações quando $q = 0,01$ e $\epsilon \geq 0,1$ ou quando $q = 0,1$ e $\epsilon \geq 0,2$, a probabilidade de erro é menor para $\tilde{\lambda}_0$ pequeno (0,25). Nestes casos temos que a probabilidade do erro converge para um valor menor do que ϵ ;

- quando $q \leq \epsilon$, pelos gráficos 6.1, 6.6 e 6.12, podemos notar que o valor de $\tilde{\lambda}_0$ não tem influência na probabilidade de erro do estimador;
- quando $\tilde{\lambda}_0$ é grande a probabilidade de erro do estimador da moda converge rapidamente para ϵ . Para valores de $q \geq 0,1$ essa convergência ocorre até o décimo passo;
- quando $\tilde{\lambda}_0$ é pequeno a moda parece sempre diminuir a probabilidade de erro, apesar de que nesses casos há maior variabilidade entre as amostras e menor estabilidade no decorrer das repetições de uma mesma amostra.

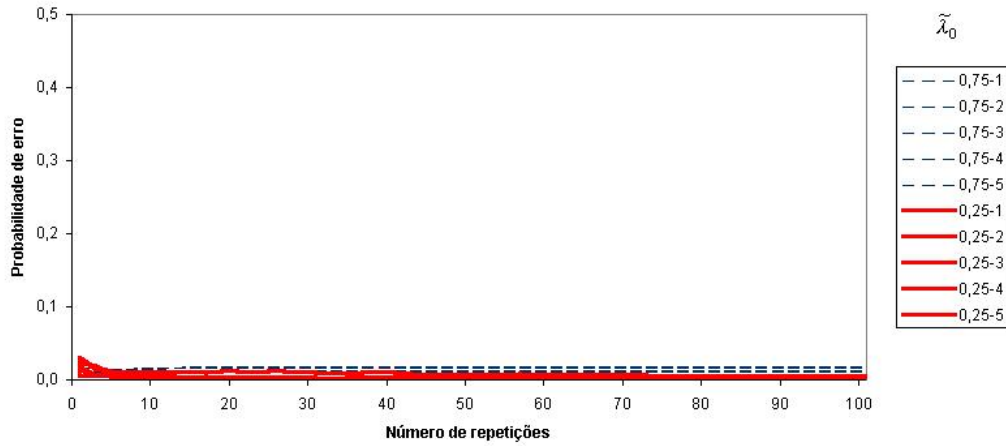


Figura 6.1: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,01$ e $\epsilon = 0,01$.

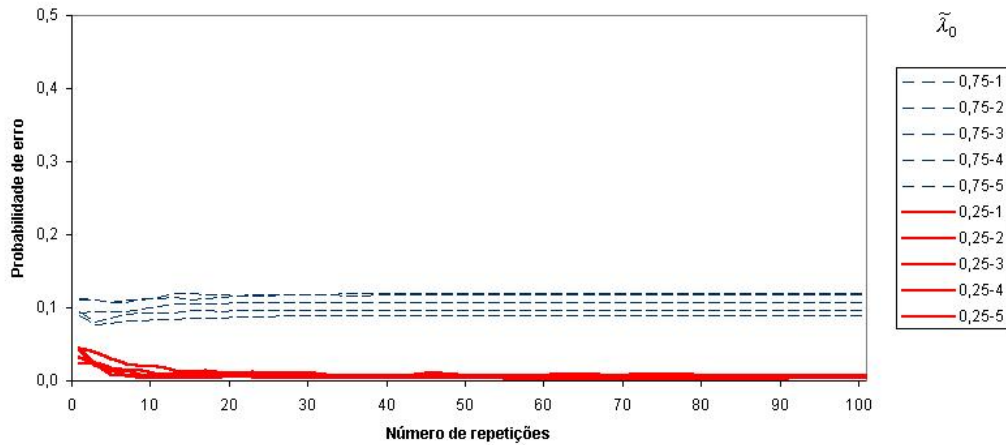


Figura 6.2: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,01$ e $\epsilon = 0,1$.

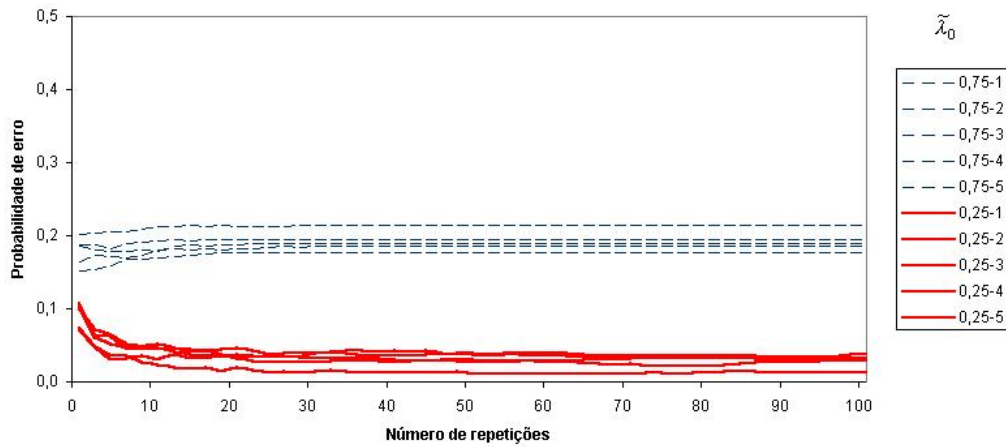


Figura 6.3: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,01$ e $\epsilon = 0,2$.

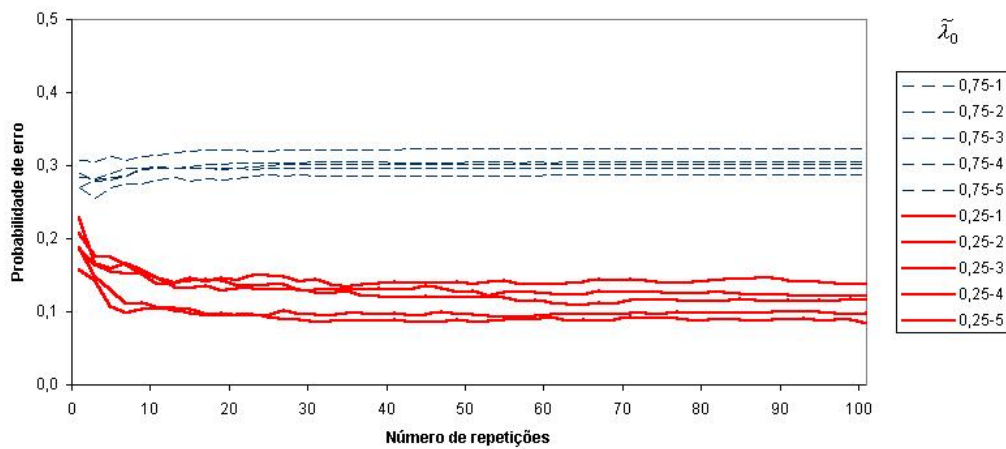


Figura 6.4: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,01$ e $\epsilon = 0,3$.

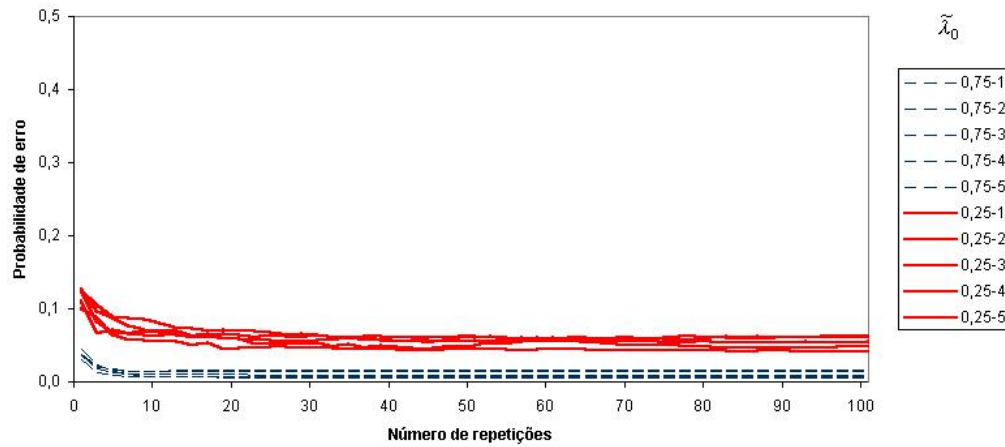


Figura 6.5: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,1$ e $\epsilon = 0,01$.

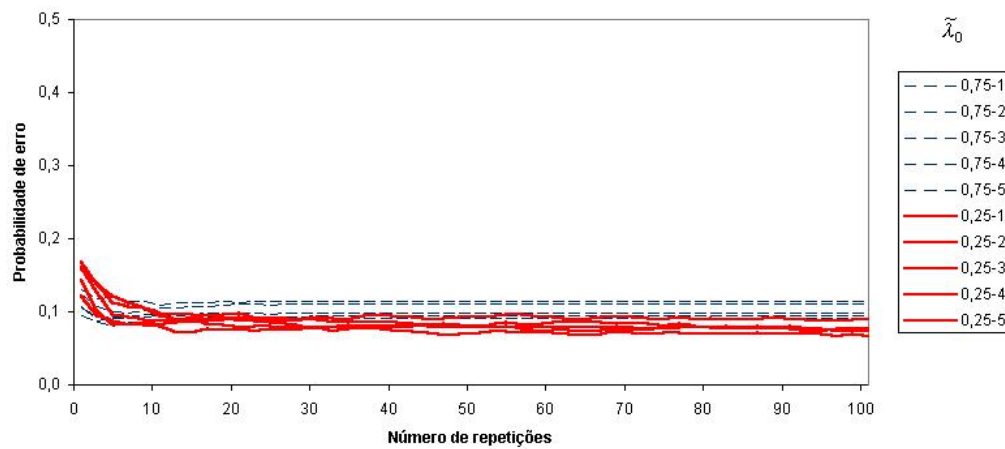


Figura 6.6: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,1$ e $\epsilon = 0,1$.

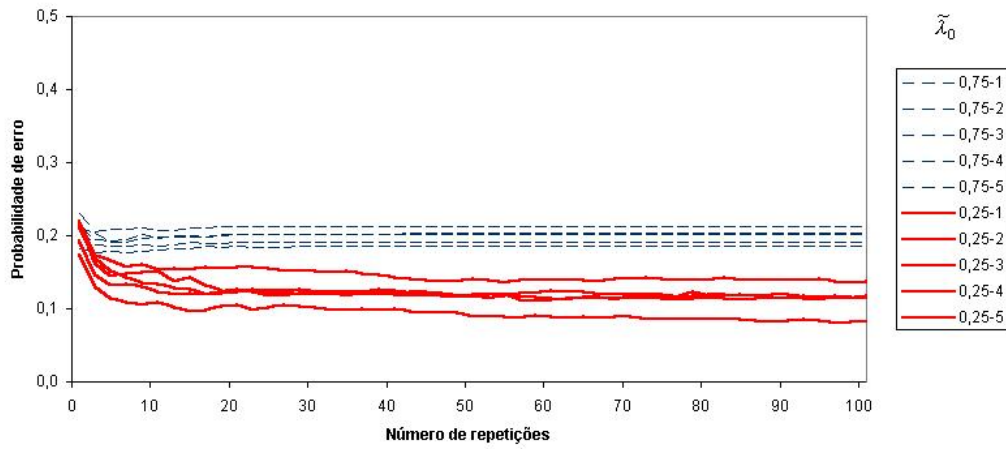


Figura 6.7: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0, 1$ e $\epsilon = 0, 2$.

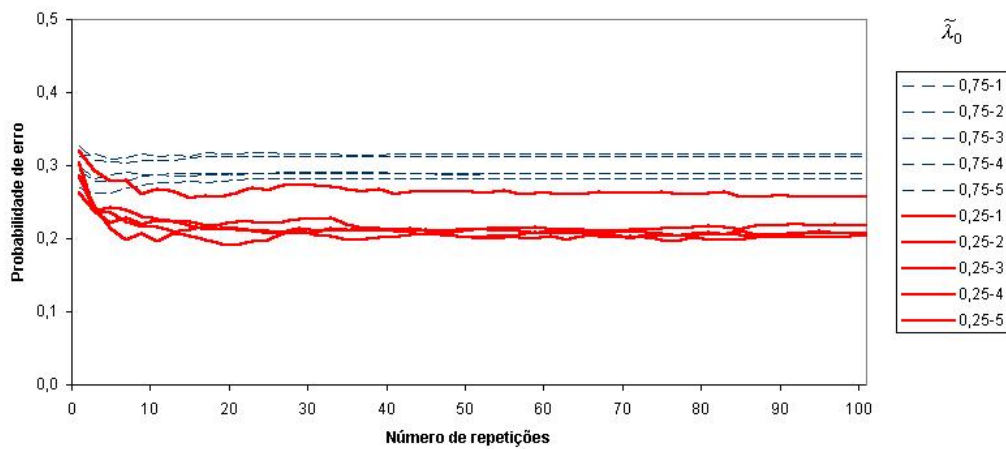


Figura 6.8: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0, 1$ e $\epsilon = 0, 3$.

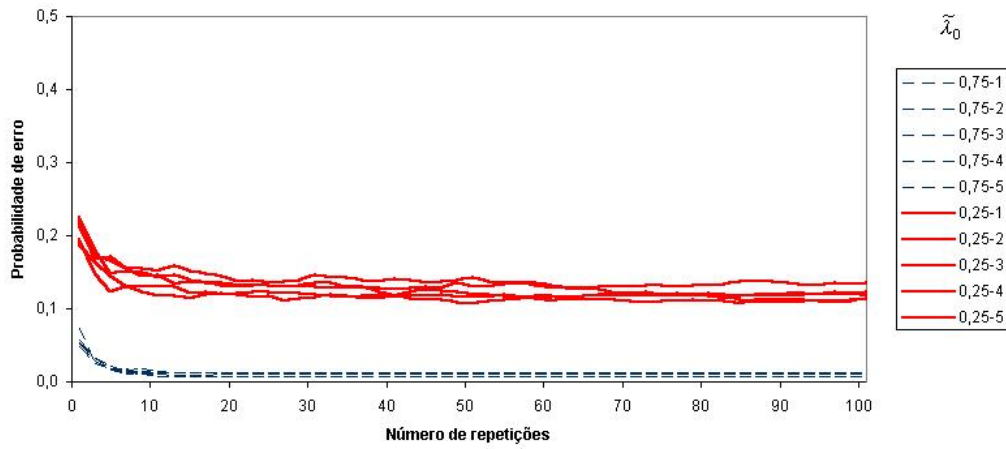


Figura 6.9: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,2$ e $\epsilon = 0,01$.

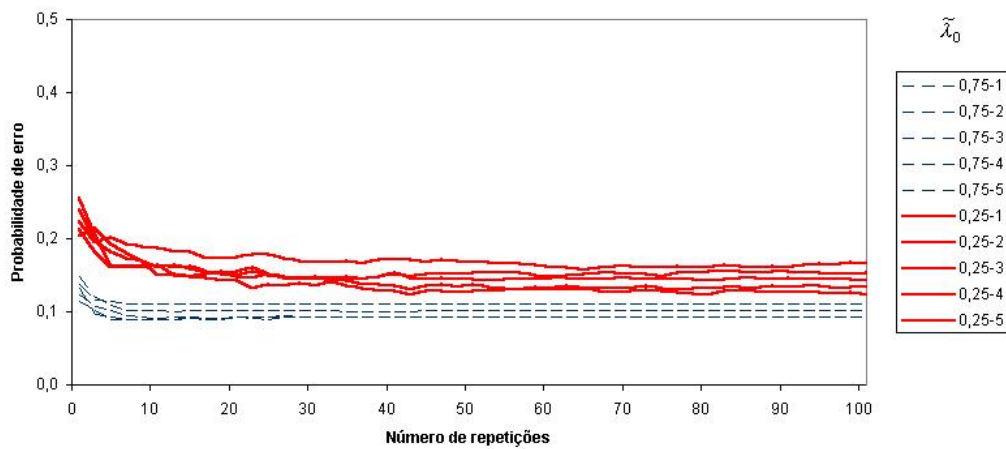


Figura 6.10: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,2$ e $\epsilon = 0,1$.

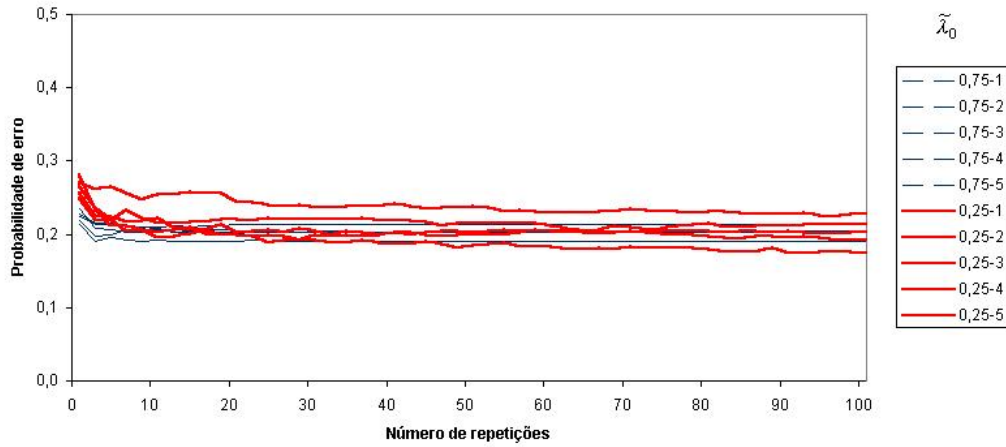


Figura 6.11: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,2$ e $\epsilon = 0,2$.

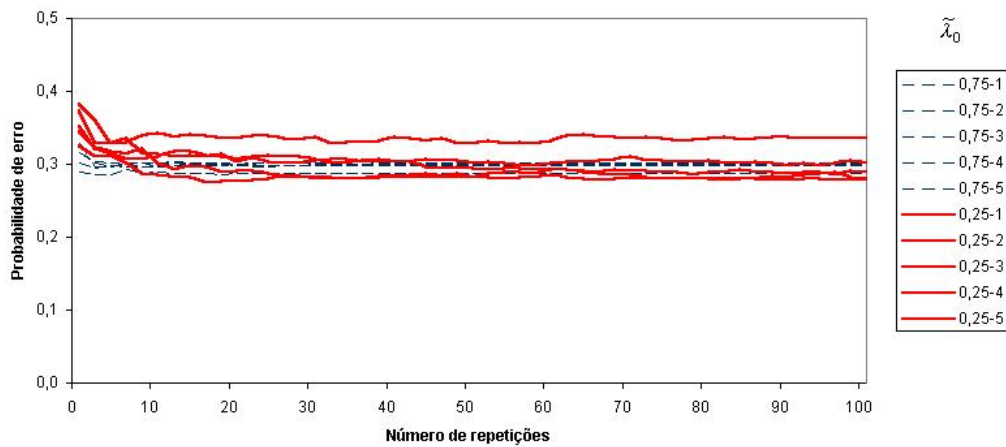


Figura 6.12: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,2$ e $\epsilon = 0,3$.

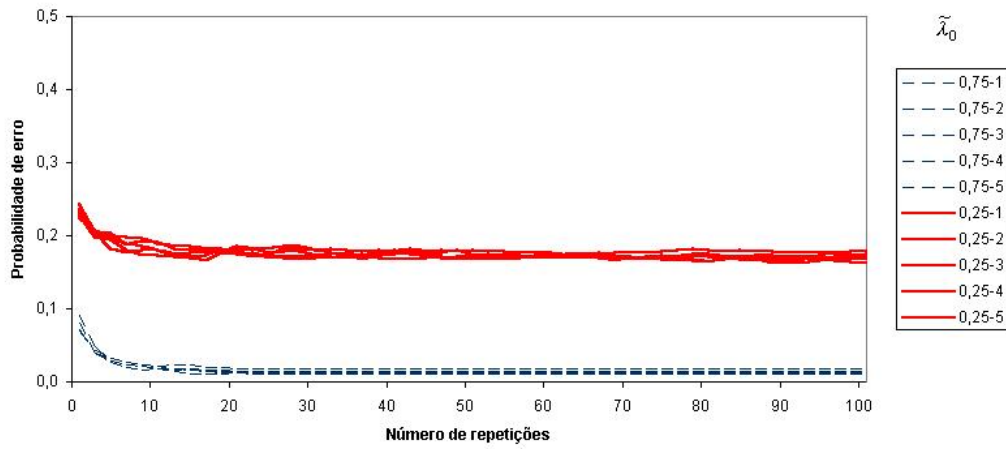


Figura 6.13: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,3$ e $\epsilon = 0,01$.

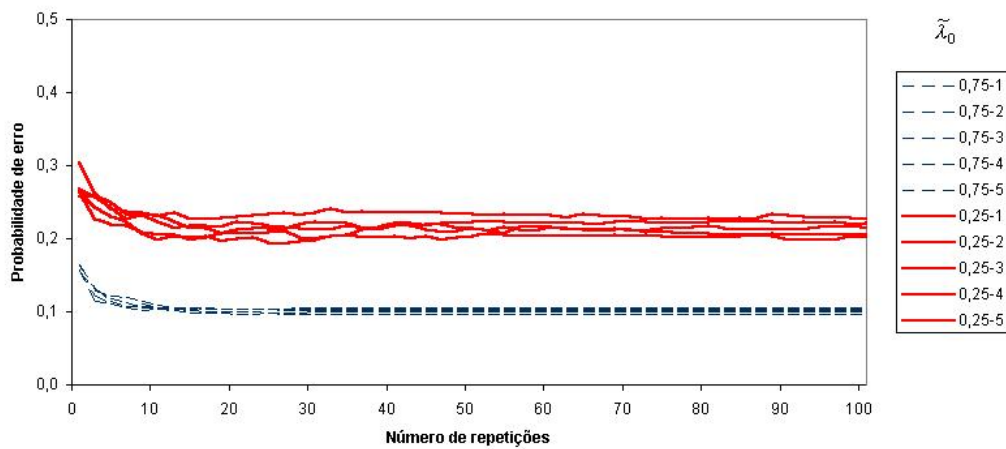


Figura 6.14: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,3$ e $\epsilon = 0,1$.

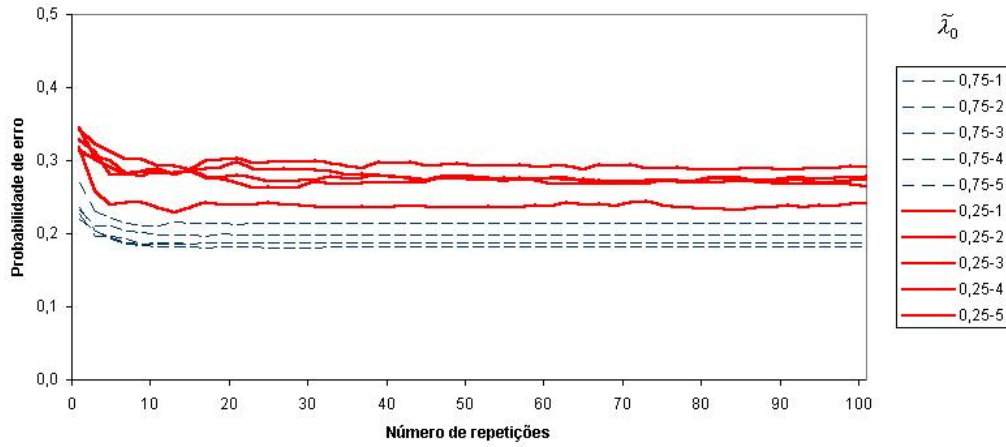


Figura 6.15: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,3$ e $\epsilon = 0,2$.

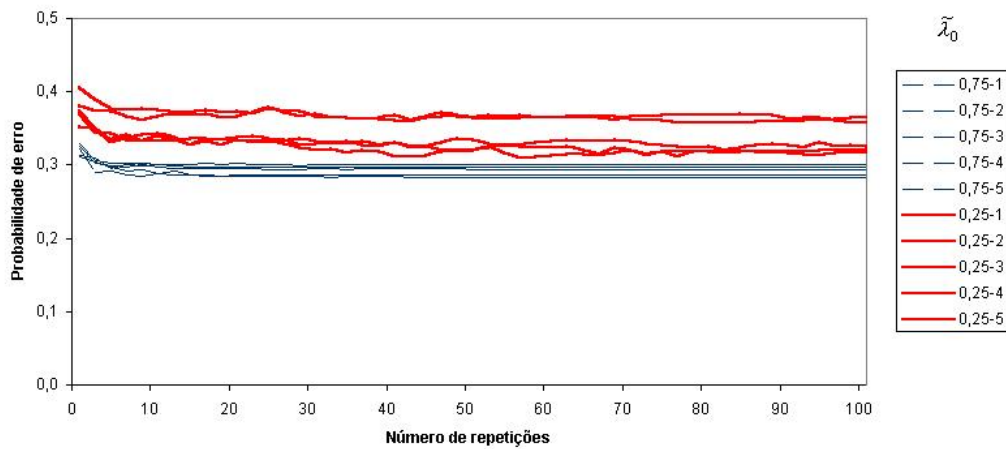


Figura 6.16: Gráfico da probabilidade de erro de \hat{X}_t com $q = 0,3$ e $\epsilon = 0,3$.

6.2 Simulações para o estimador $\bar{X}_0^T(v_0^T)$

Para avaliar a qualidade do estimador $\bar{X}_0^T(v_0^T)$ foram realizadas algumas simulações para medir a probabilidade de erro.

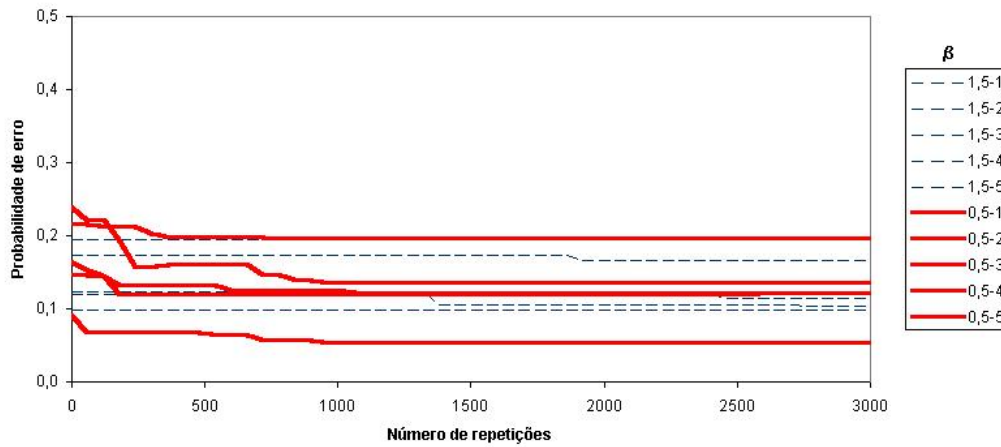
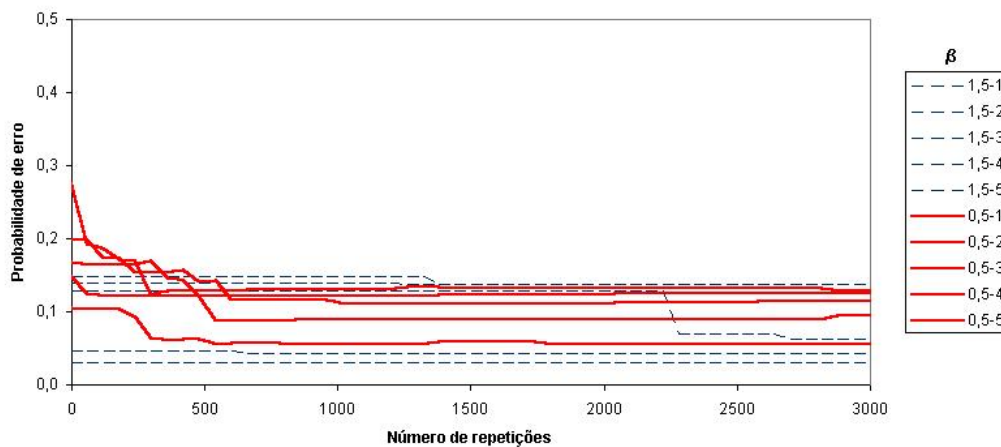
A seguir mostramos gráficos que ilustram a probabilidade de erro para alguns valores de β (0,5; 1,5), q (0,01, 0,1, 0,2, 0,3) e ϵ (0,01, 0,1, 0,2, 0,3).

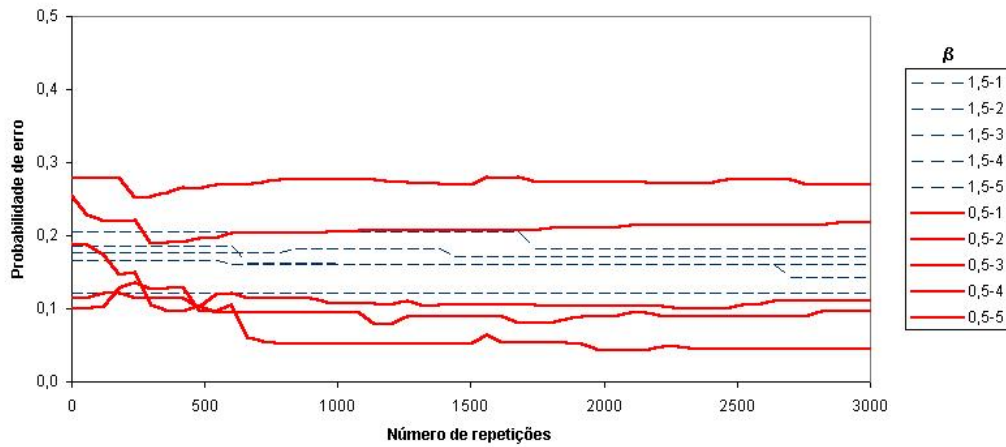
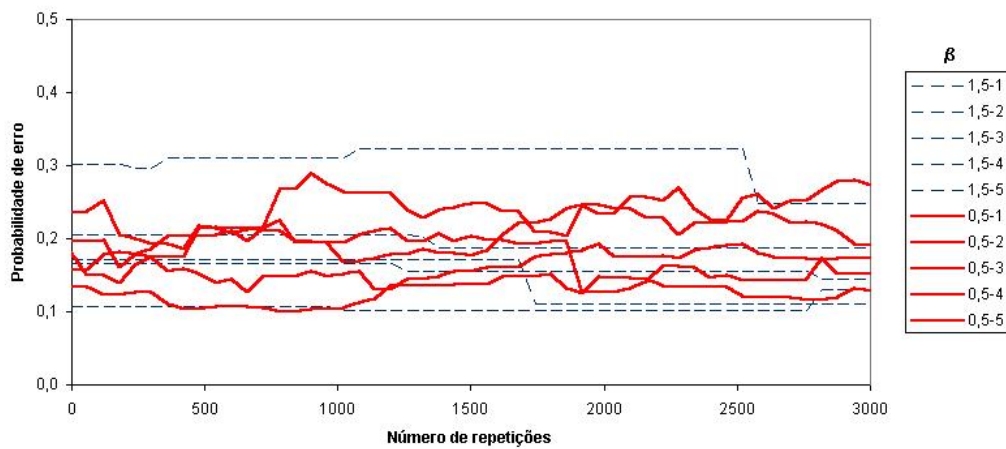
As simulações foram feitas para uma seqüência de tamanho 1000 e o número de iterações da cadeia $(\xi(k))_{k \geq 1}$ foi igual a 3000.

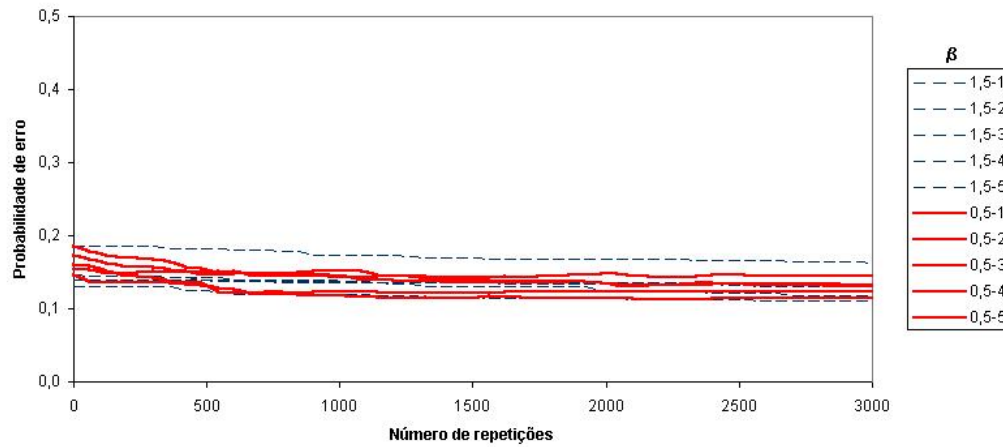
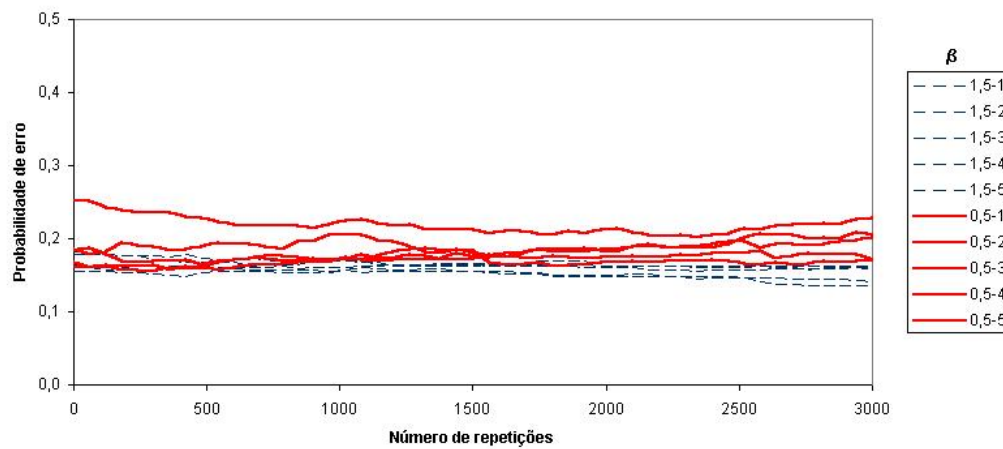
O procedimento foi repetido 5 vezes para cada combinação de β , q e ϵ .

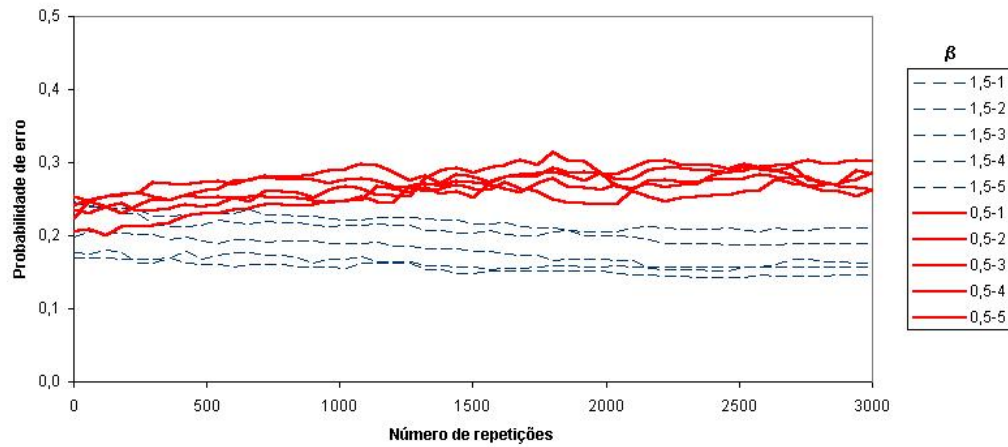
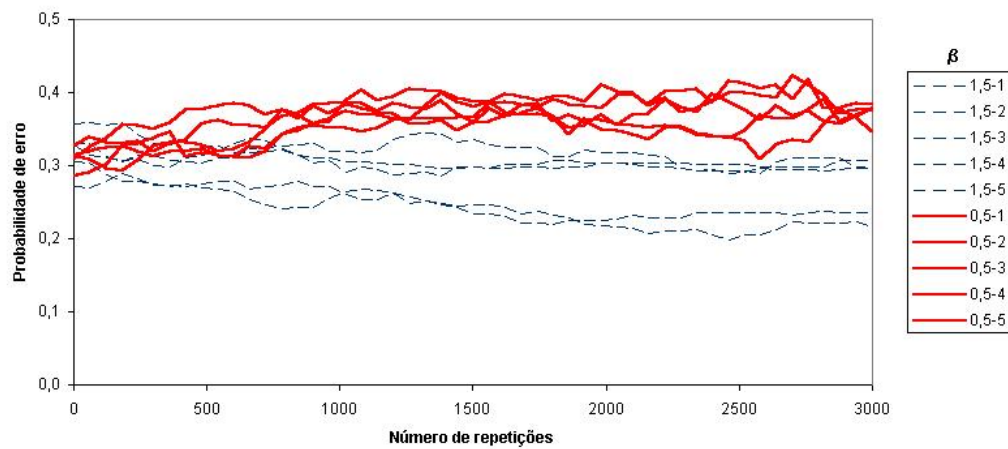
Através dos gráficos podemos tirar algumas conclusões:

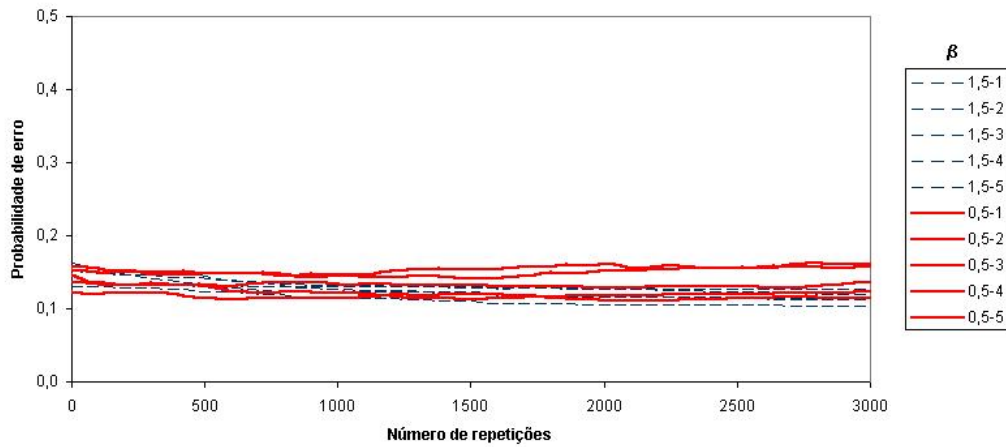
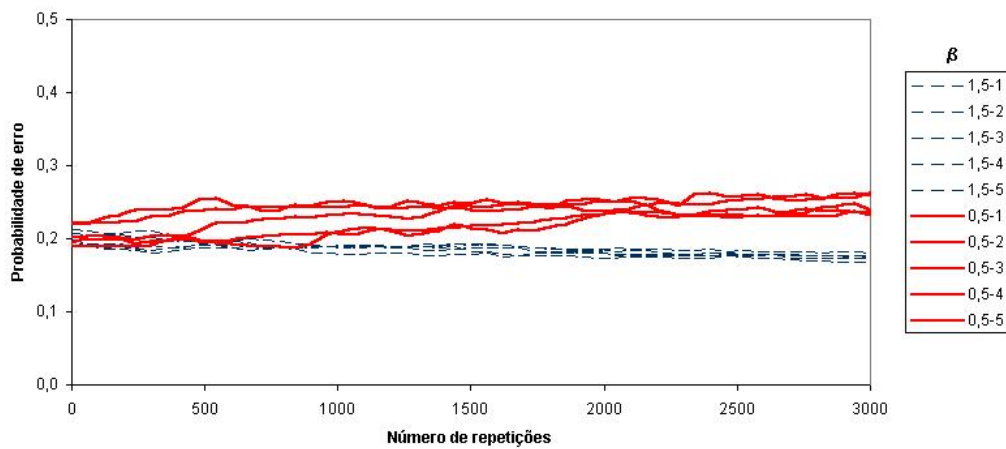
- em geral a probabilidade de erro converge para valores menores quando $\beta = 1,5$, ou seja, quando a chance de aceitarmos uma transição menos provável é menor.
- em muitos casos quando $\beta = 0,5$ notamos que a probabilidade do erro cresce à medida que aumentamos o número de passos da cadeia (ver gráficos 6.23, 6.24, 6.26, 6.27, 6.30 e 6.31);
- quando q ou ϵ são muito pequenos, pelos gráficos 6.17, 6.18, 6.19, 6.20 e 6.21, podemos notar que o valor de β não tem influência na probabilidade de erro do estimador. A exceção parece ser quando $q = 0,3$ e $\epsilon = 0,01$, pois neste caso (ver gráfico 6.25) parece que a probabilidade de erro converge para um valor menor quando $\beta = 1,5$;
- comparando com os gráficos do estimador \hat{X}_t esse estimador traz sempre resultados inferiores.

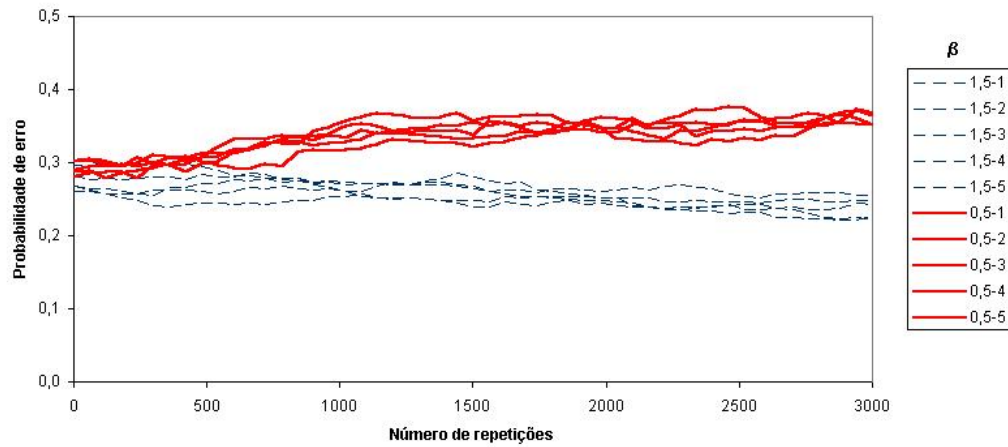
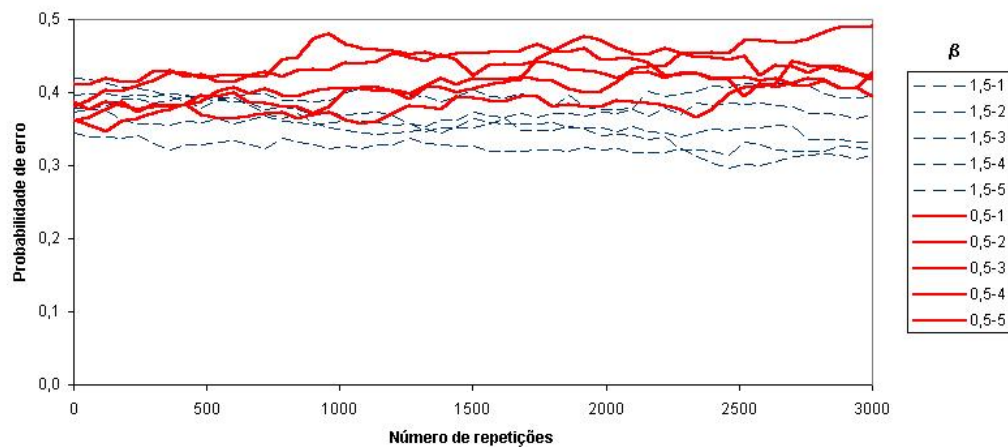
Figura 6.17: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,01$ e $\epsilon = 0,01$.Figura 6.18: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,01$ e $\epsilon = 0,1$.

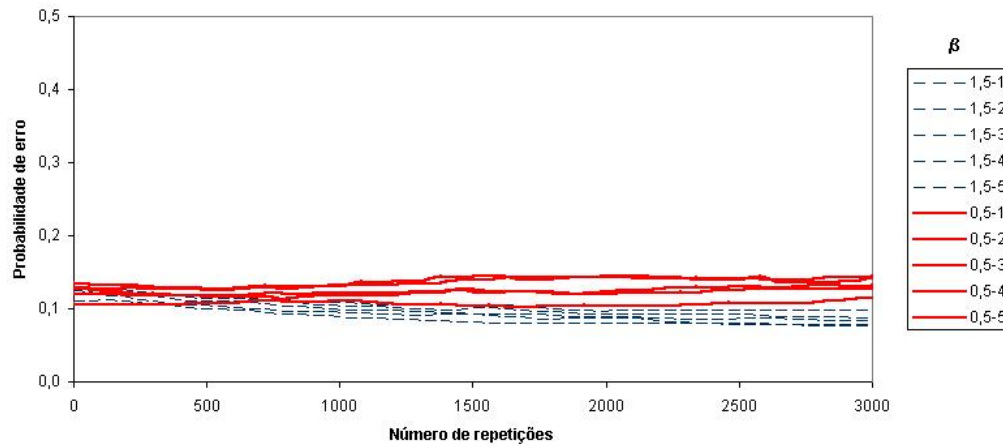
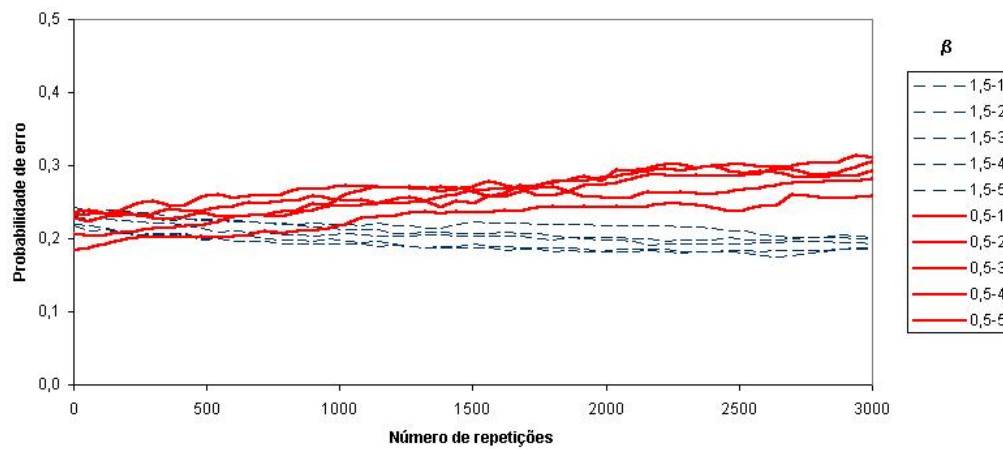
Figura 6.19: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,01$ e $\epsilon = 0,2$.Figura 6.20: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,01$ e $\epsilon = 0,3$.

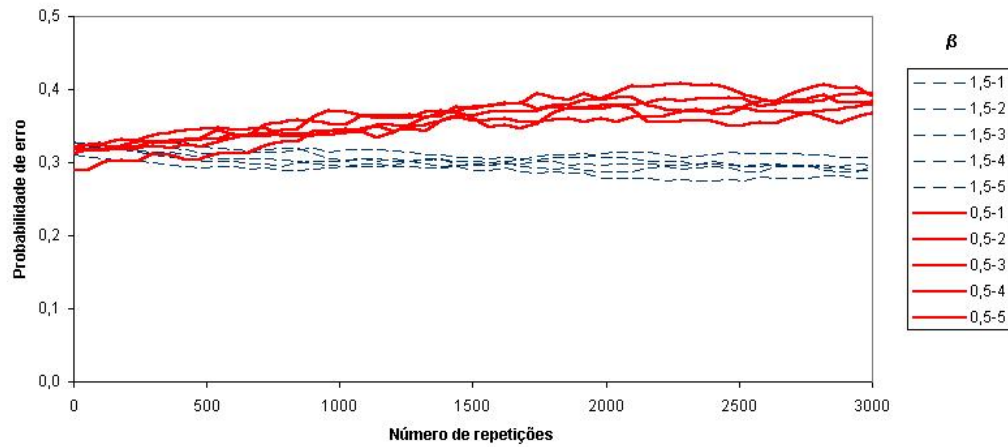
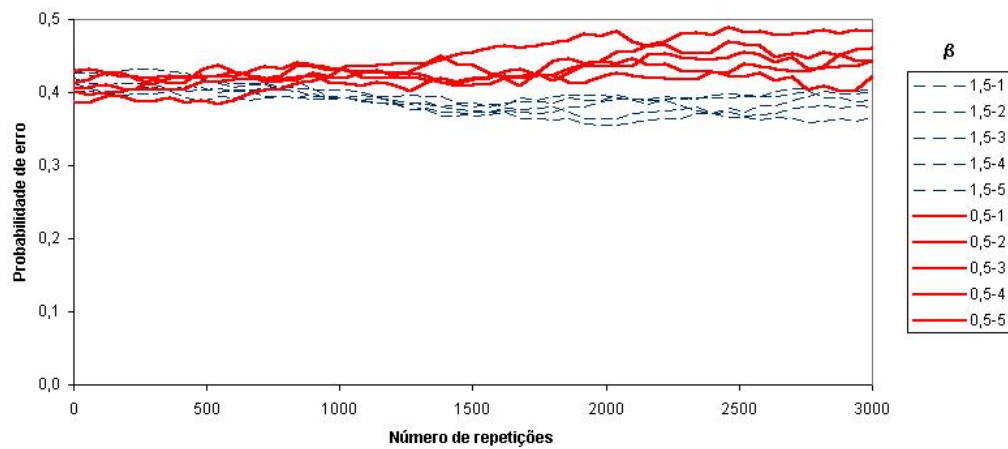
Figura 6.21: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,1$ e $\epsilon = 0,01$.Figura 6.22: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,1$ e $\epsilon = 0,1$.

Figura 6.23: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,1$ e $\epsilon = 0,2$.Figura 6.24: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,1$ e $\epsilon = 0,3$.

Figura 6.25: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,2$ e $\epsilon = 0,01$.Figura 6.26: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,2$ e $\epsilon = 0,1$.

Figura 6.27: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,2$ e $\epsilon = 0,2$.Figura 6.28: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,2$ e $\epsilon = 0,3$.

Figura 6.29: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,3$ e $\epsilon = 0,01$.Figura 6.30: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,3$ e $\epsilon = 0,1$.

Figura 6.31: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,3$ e $\epsilon = 0,2$.Figura 6.32: Gráfico da probabilidade de erro de \bar{X}_t com $q = 0,3$ e $\epsilon = 0,3$.

Apêndice A

Método de Chernoff e desigualdade de Hoeffding

Os dois resultados aqui apresentados, bem como parte de suas demonstrações foram obtidos em [2].

Método de Chernoff

Para $t > 0$, encontre o valor de s que minimize

$$P(X \geq t) = P(e^{sX} \geq e^{st}) \leq \frac{E(e^{sX})}{e^{st}}.$$

Lema A.1 *Se $E(X) = 0$ e $a \leq X \leq b$, então para qualquer $s > 0$,*

$$E(e^{sX}) \leq e^{s^2(b-a)^2/8}.$$

Prova: Como $a \leq X \leq b$, vamos definir $\alpha \in [0; 1]$ por

$$\alpha = \frac{b - X}{b - a}.$$

Note que para qualquer $s > 0$ temos,

$$\begin{aligned} s\alpha &= s \frac{b-X}{b-a} \\ s\alpha &= \frac{s(b-X)}{b-a} \\ \alpha sb - \alpha sa &= sb - sX \\ sX &= \alpha sa + (1-\alpha)sb. \end{aligned}$$

Logo, a convexidade da função exponencial implica que,

$$e^{sX} \leq \frac{b-X}{b-a} e^{sa} + \frac{X-a}{b-a} e^{sb}.$$

Calculando a esperança e tomando $p = -a/(b-a)$, podemos escrever

$$\begin{aligned} E(e^{sX}) &\leq \frac{b}{b-a} e^{sa} - \frac{a}{b-a} e^{sb} \\ &= (1-p)e^{sa} + pe^{sb} \\ &= (1-p + pe^{s(b-a)}) e^{-ps(b-a)} \\ &= e^{\phi(u)}, \end{aligned}$$

onde $u = s(b-a)$ e

$$\phi(u) = -pu + \log(1-p + pe^u).$$

Mas, fazendo a expansão de $\phi(u)$ em torno do zero, temos que

$$\phi(u) = \phi(0) + u\phi'(0) + \frac{1}{2}u^2\phi''(u^*),$$

onde $u^* \in [0; u]$ existe pelo Teorema de Taylor. Agora note que $\phi(0) = \phi'(0) = 0$ e

$$\phi''(u) = \frac{p(1-p)e^{-u}}{(p+(1-p)e^{-u})^2} \leq \frac{1}{4},$$

pois $\phi''(u)$ é apenas um caso especial de $z(1-z) \leq 1/4$ para $z = p/(p+(1-p)e^{-u})$ com $0 < z < 1$, pois $E(X) = 0$ implica que $p \in [0; 1]$. Logo,

$$E(e^{sX}) \leq e^{u^2/8} = e^{s^2(b-a)^2/8}.$$

■

Desigualdade de Hoeffding

Sejam W_1, W_2, \dots, W_n variáveis aleatórias independentes com $a_i \leq W_i \leq b_i$, e $Z_i = W_i - E(W_i)$, então

$$P\left(\left|\sum_{i=1}^n Z_i\right| \geq n\delta\right) \leq 2e^{-2n^2\delta^2/\sum_{i=1}^n (a_i-b_i)^2}.$$

Prova: Temos, pela independência das variáveis, pelo método de Chernoff e pelo lema A.1, que para todo $s > 0$

$$\begin{aligned} P\left(\left|\sum_{i=1}^n Z_i\right| \geq n\delta\right) &\leq P\left(\sum_{i=1}^n Z_i \geq n\delta\right) + P\left(\sum_{i=1}^n -Z_i \geq n\delta\right) \\ &= P\left(e^{s\sum_{i=1}^n Z_i} \geq e^{sn\delta}\right) + P\left(e^{s\sum_{i=1}^n -Z_i} \geq e^{sn\delta}\right) \\ &\leq \frac{E\left(e^{s\sum_{i=1}^n Z_i}\right)}{e^{sn\delta}} + \frac{E\left(e^{-s\sum_{i=1}^n Z_i}\right)}{e^{sn\delta}} \\ &\leq 2e^{-sn\delta} \prod_{i=1}^n e^{s^2(b_i-a_i)^2/8} \\ &= 2 \exp\left\{\frac{s^2}{8} \sum_{i=1}^n (b_i - a_i)^2 - st\right\}. \end{aligned}$$

Essa majoração é otimizada quando $s = 4n\delta / \sum_{i=1}^n (b_i - a_i)^2$, de onde concluimos a demonstração. ■

Apêndice B

Rotinas

Os algoritmos aqui apresentados contêm tanto as rotinas de simulação quanto de estimação, logo para usá-los apenas para estimar uma seqüência oculta basta omitir a parte que simula a cadeia de Markov oculta.

Os códigos apresentados neste apêndice foram feitos em R e podem ser encontrados em www.ime.usp.br/~camey/Tese.

B.1 Programa Xhat.r

Este programa executa as seguintes tarefas:

- simula uma cadeia de Markov oculta com os seguintes parâmetros: A , B e π .
- calcula $\hat{X}_t^{(j)}$ para a cadeia de Markov simulada.
- $\hat{X}_t^{(j)}$ é calculado com base em uma única seqüência estimada ($\hat{X}_t = \hat{X}_t^{(1)}$) ou com base na moda de j seqüências estimadas ($\hat{X}_t =$

$$\text{moda} \left(\hat{X}_t^{(1)}, \hat{X}_t^{(2)}, \dots, \hat{X}_t^{(j)} \right), j = 1, \dots, K.$$

```

1 library("MASS", character.only=TRUE) #carrega o pacote MASS
2 library("nnet", character.only=TRUE) #carrega o pacote nnet
3 source("C:/WINDOWS/Profiles/sucamey/My Documents/tese/rotinas
4 /funcoes2004v3.r") #carrega o arquivo de funções
5
6 Dataini<-date() #armazena o horário de início da rotina
7 print(Dataini)
8
9 N<-2 #número de estados ocultos
10 M<-2 #número de estados visíveis
11 V<-0 #vetor de estados visíveis
12 X<-0 #vetor de estados ocultos
13 A <- matrix(0,N,N) #matriz de transição da cadeia oculta
14 B <- matrix(0,N,M) #matriz de probabilidades dos estados
15 #visíveis
16
17 pi<- matrix(0,1,N) #vetor da distribuição estacionária
18 q<-0.01 #p(y|x)=q, x diferente de y
19 e<-0.01 #m_x(v)=e, x diferente de v
20 pi[1]<-0.5
21 pi[2]<-0.5
22 A[1,1]<-1-q
23 A[1,2]<-q
24 A[2,1]<-q
25 A[2,2]<-1-q
26 B[1,1]<-1-e
27 B[1,2]<-e
28 B[2,1]<-e
29 B[2,2]<-1-e
30
31
32 K<-101 #número máximo de estimativas
33 #da seqüência oculta para
34 #encontrar a moda
35
36 Fim<-1000 #Comprimento da seqüência
37
38 errmoda<-0 #vetor com erros da moda

```

```
39
40 Z<-simulaHMM2x2(Fim,A,B,pi)      #se o objetivo for somente
41                                  #estimação esse comando deve
42                                  #ser substituído pelo de leitura
43                                  #da seqüência visível
44 V<-Z[,1]
45 X<-Z[,2]
46 contador<-1
47
48 lambda0<- 0.25                    #P(Lt=0)
49 Q<-array(0,c(K,Fim))             #matriz de vetores ocultos
50 L<-array(0,c(K,Fim))             #matriz de vetores de marcas
51
52 #aqui começa a rotina de estimação
53 while (contador <= K){
54     L[contador,]<-sorteiaUm(lambda0)      #sorteia um
55                                           #vetor de marcas
56
57     Zeromax<-achaini(Fim+1,L[contador,])  #acha o início do
58                                           #último bloco
59
60     Q[contador,]<-escolhe(V,L[contador,],Zeromax)
61                                           #escolhe a seqüência
62                                           #oculta
63     Qaux<-array(0,c(contador,Fim))      #matriz auxiliar dos vetores
64                                           #ocultos
65     for (j in 1:contador){
66         Qaux[j,]<-Q[j,]
67     }
68     if (trunc(contador/2)!=(contador/2)){
69         MODA<-0
70         MODA<-moda(Qaux,contador)#calcula
71                                     #moda{Xt(1),...,Xt(contador)}
72
73         errmoda[contador/2+1]<-sum(abs(MODA-X))
74     }
75     contador<-contador+1
76 }
77 print("Erro da moda")
78 print(errmoda)
```

```

79 Datafim<-date()          #armazena o horário de fim da rotina
80 print(Datafim)
81

```

B.2 Programa Xbarra.r

Este programa executa as seguintes tarefas:

- simula uma cadeia de Markov oculta com os seguintes parâmetros: A, B e pi.
- Através do método de Monte Carlo markoviano e o algoritmo de Metrópolis calcula \bar{X}_t .

Como estado inicial o algoritmo utiliza um vetor de marcas $l_0^T(1)$ escolhido ao acaso e o vetor de estados ocultos $x_0^T(1)$ que maximiza

$$P_\theta (X_0^T(1) = x_0^T(1), L_0^T(1) = l_0^T(1) | V_0^T = v_0^T).$$

```

1  library("MASS", character.only=TRUE) #carrega o pacote MASS
2  library("nnet", character.only=TRUE) #carrega o pacote nnet
3  source("C:/WINDOWS/Profiles/sucamey/My Documents/tese/rotinas
4  /funcoes2004v2.r")          #carrega o arquivo de funções
5
6  Dataini<-date()    #armazena o horário de início da rotina
7  print(Dataini)
8  library("MASS", character.only=TRUE)
9
10 beta<-0.5    #parâmetro
11 Fim<-100     #Comprimento da seqüência
12 Total<-200   #número total de passos da cadeia
13 Q<-array(0,c(Total,Fim)) #matriz de vetores ocultos
14 L<-array(0,c(Total,Fim)) #matriz de vetores de marcas
15 contador<-2
16 N<-2        #número de estados ocultos
17 M<-2        #número de estados visíveis
18 O<-0        #vetor de estados visíveis
19 q<-0        #vetor de estados ocultos
20 A <- matrix(0,N,N) #matriz de transição da cadeia oculta
21 B <- matrix(0,N,M) #matriz de probabilidades dos estados visíveis
22 pi<- matrix(0,1,N) #vetor da distribuição estacionária

```

```
23  pistar<-0      #p0
24  Atil<-matrix(0,N,N)  #p1
25
26  p<-0.01
27  e<-0.01
28  pi[1]<-0.5
29  pi[2]<-0.5
30  A[1,1]<-1-p
31  A[1,2]<-p
32  A[2,1]<-p
33  A[2,2]<-1-p
34  B[1,1]<-1-e
35  B[1,2]<-e
36  B[2,1]<-e
37  B[2,2]<-1-e
38
39  Z<-simulaHMM2x2(Fim,A,B,pi)
40  0<-Z[,1]
41  q<-Z[,2]
42
43  alfastar<-calculaalfa(A) #lambda0
44  pistar<-calculapistar(A,alfastar)
45  Atil<-calculaAtil(A)
46
47
48  L[1,]<-sorteiaUm(alfastar)
49  Zeromax<-achaini(Fim+1,L[1,]) #determina max{t:L_t=0}
50  Q[1,]<-escolhe(0,Atil,B,pi,pistar,L[1,],Zeromax)
51
52  while (contador<Total+1){
53    UD<-unifdisc(Fim) #uniforme discreta
54    U<-runif(1)
55    L[contador,]<-L[contador-1,]
56    if (U<=((alfastar)^beta)/(alfastar^beta+(1-alfastar)^beta))
57      L[contador,UD]<-0
58    if (U>(((alfastar)^beta)/(alfastar^beta+(1-alfastar)^beta))
59      L[contador,UD]<-1
60    Q[contador,]<-Q[contador-1,]
61    if (L[contador,UD]!=L[contador-1,UD]) {
62      if (UD==1){
```

```

63     if (L[contador,UD]==0){
64         inibloco<-1
65         fimbloco<-achafim(UD, L[contador,])
66         lbloco<-fimbloco-1
67         W<-bloco(inibloco,fimbloco,Q[contador-1,])
68         for (j in 1:lbloco){
69             Q[contador,j]<-W[j]
70         }
71         R<-(MLbloco(Q[contador,],pi,1,fimbloco)
72             /(MLbloco(Q[contador-1,],pi,1,fimbloco)))^beta
73         if (R<1){
74             P<-runif(1)
75             if (P>R) {
76                 Q[contador,]<-Q[(contador-1),]
77                 L[contador,]<-L[(contador-1),]
78             }
79         }
80     }
81 }
82 if (UD!=1){
83     if (((L[contador,UD]==1) | (UD==Fim)) &
84         (Q[contador-1,UD]!=Q[contador-1,UD-1])){
85         inibloco<-achaini(UD, L[contador,])
86         fimbloco<-achafim(UD, L[contador,])
87         lbloco<-fimbloco-inibloco
88         W<-bloco(inibloco,fimbloco,Q[contador-1,])
89         for (j in 1:lbloco){
90             Q[contador,inibloco+j-1]<-W[j]
91         }
92         if (inibloco==1){
93             R<-(MLbloco(Q[contador,],pi,1,fimbloco)/
94                 (MLbloco(Q[contador-1,],pi,1,UD)*
95                 MLbloco(Q[contador-1,],pistar,UD,fimbloco)))^beta
96         }
97         if (inibloco!=1){
98             R<-(MLbloco(Q[contador,],pistar,Q[contador-1,],inibloco,fimbloco)/
99                 (MLbloco(Q[contador-1,],pistar,inibloco,UD)*
100                 MLbloco(Q[contador-1,],pistar,UD,fimbloco)))^beta
101         }
102         if (R<1){

```



```

103         P<-runif(1)
104         if (P>R) {
105             Q[contador,]<-Q[(contador-1),]
106             L[contador,]<-L[(contador-1),]
107         }
108     }
109 }
110 if ((L[contador,UD]==0) & (UD!=Fim)){
111     inibloco<-achaini(UD, L[contador,])
112     fimbloco<-achafim(UD, L[contador,])
113     W<-bloco(UD,fimbloco,Q[contador-1,])
114     for (j in 1:(fimbloco-UD)){
115         Q[contador,UD+j-1]<-W[j]
116     }
117     if (inibloco==1){
118         R<-(MLbloco(Q[contador,],pi,1,UD)*
119             MLbloco(Q[contador,],pistar,UD,fimbloco)/
120             MLbloco(Q[contador-1,],pi,1,fimbloco))^beta
121     }
122     if (inibloco!=1){
123         R<-(MLbloco(Q[contador,],pistar,inibloco,UD)*
124             MLbloco(Q[contador,],pistar,UD,fimbloco)/
125             MLbloco(Q[contador-1,],pistar,inibloco,fimbloco))^beta
126     }
127     if (R<1){
128         P<-runif(1)
129         if (P>R) {
130             Q[contador,]<-Q[(contador-1),]
131             L[contador,]<-L[(contador-1),]
132         }
133     }
134 }
135 }
136 }
137 contador<-contador+1
138 }
139
140 dif<-matrix(0>Total,Fim)
141 erros<-0
142 for (i in 1>Total){

```

```
143   dif[i,]<-abs(Q[i,]-q)
144   erros[i]<-sum(dif[i,])
145 }
146 print("Erros")
147 print(erros)
148
149 Datafim<-date()   #armazena o horário de fim da rotina
150 print(Datafim)
151
```

B.3 Programa funcoes2004v2.r

Este programa contém as funções utilizadas no programa Xbarra.r.

```
1  # 1 para determinar o valor de lambda0 para uma determinada matriz A.
2  # 2 simula uma Bernoulli com lambda0 e guardar as posições das marcas.
3  # 3 para calcular p0 referente à A e lambda0.
4  # 4 para calcular a matriz p1 referente a A, p0 e lambda0.
5  # 5 constrói os vetores ocultos.
6  # 6 calcula verossimilhança dos blocos, mas o vetor oculto é só do
7  #   bloco.
8  # 7 calcula verossimilhança dos blocos, mas o vetor oculto é toda
9  #   a seqüência.
10 # 8 escolhe seqüência oculta de acordo com o critério de consenso.
11 # 9 escolhe bloco oculto.
12 #10 gera uniforme discreta.
13 #11 encontra inicio do bloco.
14 #12 encontra fim do bloco.
15 #13 simula uma HMM 2x2.
16
17 #1
18
19 calculaalfa<-function(A1){
20   alfamin<-0
21   if (A1[1,1]>A1[2,1])
22     alfamin<- A1[2,1]+A1[1,2]
23   if (A1[1,1]<A1[2,1])
24     alfamin<- A1[1,1]+A1[2,2]
25   alfamin
26 }
```

```
27
28
29 #2
30 sorteiaUm<-function(alfamin){
31   U2<-runif(Fim, min=0, max=1)
32   eta<-0
33   eta[1]<-1
34 # eta[Fim]<-1
35   for (i in 1:Fim) {
36     if (U2[i]<=alfamin) eta[i]<-0
37     if (U2[i]>alfamin) eta[i]<-1
38   }
39   eta
40 }
41
42
43
44 #3
45 calculapistar<-function(A1,alfamin){
46   pistar<- matrix(0,1,N)
47   if (A1[1,1]>A1[2,1]){
48     pistar[1]<-A1[2,1]/alfamin
49     pistar[2]<-A1[1,2]/alfamin
50   }
51   if (A1[1,1]<A1[2,1]){
52     pistar[1]<-A1[1,1]/alfamin
53     pistar[2]<-A1[2,2]/alfamin
54   }
55   pistar
56 }
57
58 #3'
59 calculaAtil<-function(A1){
60   Atil<-matrix(0,N,N)
61   if (A1[1,1]>A1[2,1]){
62     Atil[1,1]<-1
63     Atil[2,2]<-1
64     Atil[1,2]<-0
65     Atil[2,1]<-0
66   }
```

```
67   if (A1[1,1]<A1[2,1]){
68     Atil[1,1]<-0
69     Atil[2,2]<-0
70     Atil[1,2]<-1
71     Atil[2,1]<-1
72   }
73   Atil
74 }
75
76 #5
77 vetoroculto<-function(lin,N,n){
78   q<-0
79   r<-0
80   num<-lin
81   col<-n
82   W<-matrix(1,1,n)
83   while(num>0){
84     q<-trunc(num/N)
85     r<-N-1
86     W[col]<-r+1
87     col<-col-1
88     num<-q
89   }
90   W
91 }
92
93 #6
94 verossim<-function(q,pi,inicio,fim){
95   ML<-pi[q[1]]*B[q[1],0[inicio]]
96   if ((fim-inicio-1)>0){
97     for (k in 1:(fim-inicio-1)){
98       ML<-ML*Atil[q[k],q[k+1]]*B[q[k+1],0[inicio+k]]
99     }
100  }
101  ML
102 }
103
104 #7
105 MLbloco<-function(q,pi,inicio,fim){
106   ML<-pi[q[inicio]]*B[q[inicio],0[inicio]]
```

```
107   if ((fim-inicio-1)>0){
108     for (k in 1:(fim-inicio-1)){
109       ML<-ML*Atil[q[inicio+k-1],q[inicio+k]]*
110         B[q[inicio+k],0[inicio+k]]
111     }
112   }
113   ML
114 }
115
116 #8
117 escolhe<-function(O,A,B,pi,pistar,L,Zeromax){
118   qescolhido<-0
119   U<-0
120   Qescolhida<-0
121
122   #escolhe o primeiro bloco da seq. oculta
123   inicio<-1
124   fim<-achafim(1,L) # acha o fim do 1o. bloco
125   n<-fim-inicio # tamanho do 1o. bloco
126   if (fim==Fim){ # se o bloco termina no fim da seqüência
127     n<-n+1
128   }
129   Qaux<-0 # guarda a verossimilhança do 1o. bloco
130   W<-vetoroculto(0,N,n) # vetor de 1's
131   Qaux<-verossim(W,pi,inicio,fim) # calcula a verossimilhança
132   # do bloco
133   W<-vetoroculto(N^n-1,N,n) # vetor de 2's
134   Qaux<-verossim(W,pi,inicio,fim)+Qaux
135   U<-runif(1)
136   W<-vetoroculto(0,N,n)
137   if (U<=verossim(W,pi,inicio,fim)/Qaux) qescolhido<-1 #escolhe o
138   #vetor de 1's
139   if (U>verossim(W,pi,inicio,fim)/Qaux) qescolhido<-N^n #escolhe o
140   #vetor de 2's
141   Qescolhida<-vetoroculto(qescolhido-1,N,n) #armazena em
142   #Qescolhida o 1o. bloco escolhido
143   #escolhe o resto da seqüência oculta
144   while (fim<=Zeromax){
145     inicio<-fim # início é o fim do bloco anterior
146     fim<-achafim(inicio,L) # acha o fim do bloco
```

```
147     n<-fim-inicio # tamanho do bloco
148     if (fim==Fim){ # se o bloco termina no fim da seqüência
149         n<-n+1
150     }
151     Qaux<-0
152     W<-vetoroculto(0,N,n)
153     Qaux<-verossim(W,pistar,inicio,fim)
154     W<-vetoroculto(N^n-1,N,n)
155     Qaux<-verossim(W,pistar,inicio,fim)+Qaux
156     U<-runif(1)
157     W<-vetoroculto(0,N,n)
158     if (U<=verossim(W,pistar,inicio,fim)/Qaux) qescolhido<-1
159     if (U>verossim(W,pistar,inicio,fim)/Qaux) qescolhido<-N^n
160     W<-vetoroculto(qescolhido-1,N,n)
161     for (j in 1:n){
162         Qescolhida[inicio-1+j]<-W[j]
163     }
164     if (Zeromax==Fim & fim==Fim) fim<-fim+1
165 }
166 Qescolhida
167 }
168 #9
169 bloco<-function(inicio,fim,q){
170     n<-fim-inicio
171     if (q[inicio]==1) qescolhido<-N^n
172     if (q[inicio]==2) qescolhido<-1
173     Qescolhida<-vetoroculto(qescolhido-1,N,n)
174     Qescolhida
175 }
176
177
178 #10
179 unifdisc<-function(Fim){
180     U<-0
181     U<-trunc(runif(1,min=1,max=(Fim+1)))
182     U
183 }
184
185 #11
186 achaini<-function(UD,L){
```

```
187   I<-1
188   i<-UD
189   while (i>1){
190     if (L[i-1]==0) {
191       I<-i-1
192       i<--1
193     }
194     i<-i-1
195   }
196   I
197 }
198
199 #12
200 achafim<-function(UD,L){
201   I<-Fim
202   i<-UD
203   while (i<Fim){
204     if (L[i+1]==0) {
205       I<-i+1
206       i<-Fim
207     }
208     i<-i+1
209   }
210   I
211 }
212
213 #13
214 simulaHMM2x2<- function(Fim,A,B,pi){
215   U<-runif(Fim, min=0, max=1)
216   U1<-runif(Fim, min=0, max=1)
217   if (U[1]<=pi[1]) q[1]<-1
218   if (U[1]>pi[1]) q[1]<-2
219
220   for (i in 2:Fim) {
221     if (q[i-1]==1) {
222       if (U[i]<=A[1,1]) q[i]<-1;
223       if (U[i]>A[1,1]) q[i]<-2;
224     }
225     if (q[i-1]==2) {
226       if (U[i]<=A[2,1]) q[i]<-1;
```

```
227     if (U[i]>A[2,1]) q[i]<-2;
228   }
229 }
230
231 for (i in 1:Fim) {
232   if (q[i]==1) {
233     if (U1[i]<=B[1,1]) O[i]<-1;
234     if ((U1[i]>B[1,1])) O[i]<-2;
235   }
236   if (q[i]==2) {
237     if (U1[i]<=B[2,1]) O[i]<-1;
238     if ((U1[i]>B[2,1])) O[i]<-2;
239   }
240 }
241 Z<-cbind(O,q)
242 Z
243 }
```

B.4 Programa funcoes2004v3.r

Este programa contém as funções utilizadas no programa `estimador1.r`.

```
1 #1 simula uma Bernoulli com lambda0 e guardar as posições das marcas.
2 #2 constrói os vetores ocultos.
3 #3 escolhe seqüência oculta pela maioria ou ao acaso quando tamanho
4 # do bloco é par e não existe maioria.
5 #4 encontra inicio do bloco.
6 #5 encontra fim do bloco.
7 #6 encontra a moda.
8 #7 simula uma HMM 2x2.
9
10 #1
11 sorteiaUm<-function(alfamin){
12   U2<-runif(Fim, min=0, max=1)
13   eta<-0
14   eta[1]<-1
15   for (i in 2:Fim) {
16     if (U2[i]<=alfamin) eta[i]<-0
17     if (U2[i]>alfamin) eta[i]<-1
18   }
```



```
19   eta
20 }
21
22
23 #2
24 vetoroculto<-function(lin,N,n){
25   q<-0
26   r<-0
27   num<-lin
28   col<-n
29   W<-matrix(1,1,n)
30   while(num>0){
31     q<-trunc(num/N)
32     r<-N-1
33     W[col]<-r+1
34     col<-col-1
35     num<-q
36   }
37   W
38 }
39
40 #3
41 escolhe<-function(O,L,Zeromax){
42   qescolhido<-0
43   U<-0
44   Qescolhida<-0
45
46   #escolhe o primeiro bloco da seq. oculta
47   inicio<-1
48   fim<-achafim(1,L) # acha o fim do 1o. bloco
49   n<-fim-inicio # tamanho do 1o. bloco
50   if (fim==Fim){ # se o bloco termina no fim da seqüência
51     n<-n+1
52   }
53   Qaux<-0 # guarda o número de 2's nos blocos
54   W<-vetoroculto(O,N,n) # vetor visível
55   for (j in 1:n){
56     W[j]<-O[inicio+j-1]-1 # guarda vetor visível
57   }
58   Qaux<-sum(W) # calcula o no. de 2's no bloco
```

```
59   if (Qaux<(n/2)) qescolhido<-1 #escolhe o vetor de 1's
60   if (Qaux>(n/2)) qescolhido<-N^n #escolhe o vetor de 2's
61   if (Qaux==(n/2)){
62     U<-0
63     U<-runif(1,0,1)
64     if (U<=0.5) qescolhido<-1
65     if (U>0.5) qescolhido<-N^n
66   }
67   Qescolhida<-vetoroculto(qescolhido-1,N,n) #armazena em Qescolhida
68 # o 1o. bloco
69     #escolhido
70 #escolhe o resto da seqüência oculta
71   while (fim<=Zeromax){
72     inicio<-fim # início é o fim do bloco anterior
73     fim<-achafim(inicio,L) # acha o fim do bloco
74     n<-fim-inicio # tamanho do bloco
75     if (fim==Fim){ # se o bloco termina no fim da seqüência
76       n<-n+1
77     }
78     Qaux1<-0
79     W<-vetoroculto(0,N,n)
80     for (j in 1:n){
81       W[j]<-0[inicio+j-1]-1 # guarda vetor visível
82     }
83     Qaux1<-sum(W) # calcula o no. de 2's no bloco
84     if (Qaux1<(n/2)) qescolhido<-1
85     if (Qaux1>(n/2)) qescolhido<-N^n
86     if (Qaux1==(n/2)){
87       U<-0
88       U<-runif(1,0,1)
89       if (U<=0.5) qescolhido<-1
90       if (U>0.5) qescolhido<-N^n
91     }
92     W<-vetoroculto(qescolhido-1,N,n)
93     for (j in 1:n){
94       Qescolhida[inicio-1+j]<-W[j]
95     }
96     if (Zeromax==Fim & fim==Fim) fim<-fim+1
97   }
98   Qescolhida
```

```
99  }
100
101  #4
102  achaini<-function(UD,L){
103    I<-1
104    i<-UD
105    while (i>1){
106      if (L[i-1]==0) {
107        I<-i-1
108        i<--1
109      }
110      i<-i-1
111    }
112    I
113  }
114
115  #5
116  achafim<-function(UD,L){
117    I<-Fim
118    i<-UD
119    while (i<Fim){
120      if (L[i+1]==0) {
121        I<-i+1
122        i<-Fim
123      }
124      i<-i+1
125    }
126    I
127  }
128
129  #6
130  moda<-function(q){
131    for (i in 1:Fim){
132      aux<-0
133      for (k in 1:Total){
134        aux<-q[k,i]-1+aux
135      }
136      if (aux<(Total/2)) M[i]<-1
137      if (aux>(Total/2)) M[i]<-2
138    }

```

```
139   M
140 }
141
142
143 #7
144 simulaHMM2x2<- function(Fim,A,B,pi){
145   U<-runif(Fim, min=0, max=1)
146   U1<-runif(Fim, min=0, max=1)
147   if (U[1]<=pi[1]) q[1]<-1
148   if (U[1]>pi[1]) q[1]<-2
149
150   for (i in 2:Fim) {
151     if (q[i-1]==1) {
152       if (U[i]<=A[1,1]) q[i]<-1;
153       if (U[i]>A[1,1]) q[i]<-2;
154     }
155     if (q[i-1]==2) {
156       if (U[i]<=A[2,1]) q[i]<-1;
157       if (U[i]>A[2,1]) q[i]<-2;
158     }
159   }
160
161   for (i in 1:Fim) {
162     if (q[i]==1) {
163       if (U1[i]<=B[1,1]) O[i]<-1;
164       if ((U1[i]>B[1,1])) O[i]<-2;
165     }
166     if (q[i]==2) {
167       if (U1[i]<=B[2,1]) O[i]<-1;
168       if ((U1[i]>B[2,1])) O[i]<-2;
169     }
170   }
171   Z<-cbind(O,q)
172   Z
173 }
```

Referências Bibliográficas

- [1] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Statist.*, 37:1554–1563, 1966.
- [2] Luc Devroye. Exponential inequalities in nonparametric estimation. In *Nonparametric functional estimation and related topics (Spetses, 1990)*, volume 335 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 31–44. Kluwer Acad. Publ., Dordrecht, 1991.
- [3] Yariv Ephraim and Neri Merhav. Hidden Markov processes. *IEEE Trans. Inform. Theory*, 48(6):1518–1569, 2002. Special issue on Shannon theory: perspective, trends, and applications.
- [4] R. Fernandez, P. A. Ferrari, and A. Galves. Coupling, renewal and perfect simulation of chains of infinite order, 2001. Notes for a Course in the V Brazilian School of Probability.
- [5] G. Golubev and R. Khasminskii. Asymptotically optimal filtering for a hidden Markov model. *Math. Methods Statist.*, 7(2):192–209, 1998.
- [6] Rafail Khasminskii and Ofer Zeitouni. Asymptotic filtering for finite state Markov chains. *Stochastic Process. Appl.*, 63(1):1–10, 1996.

Índice Remissivo

Símbolos	N_x	27
A	$(p^0(y))_{y \in A}$	11
$A_V(x_0^T y_0^T, l_0^T)$	$(p^1(y x))_{x, y \in A}$	11
B	$p_L(l_0^T r_0^T)$	54
B_i	Q	8
B_{fim}	q	16
B_{inicio}	$q(y x)$	8
β	R	53
$D_T^1(\hat{X}_0^T(v_0^T, \tilde{l}_0^T))$	τ^*	51
Δ_s	τ_i	16
$(\delta_t)_{t \geq 0}$	τ_i^t	27
ϵ	$\tilde{\tau}_1^t$	39
η	$\tilde{\tau}_0^t$	39
η_T	θ	8
$F_v(x_0^T, l_0^T)$	U_1	52
J_T	U_2	53
λ_0	V_t	8
$\tilde{\lambda}_0$	X_t	8
λ_1	$\bar{X}_0^T(v_0^T)$	49
$(L_t)_{t \geq 0}$	$\ddot{X}_0^T(v_0^T, l_0^T)$	26
$(\tilde{L}_t)_{t \in \mathcal{Z}}$	$\hat{X}_t(v_0^T, \tilde{l}_0^T)$	40
$l_0^T(k)$	$x_0^T(k)$	51
l_i	$(\xi(k))_{k \geq 1}$	51
$l_i(t)$	$(\xi_t)_{t \geq 0}$	17
$\tilde{l}(t)$	W_i	37
m	Y_i	36
$m_x(v)$	$Z_{v, \beta}$	50
$\mu(x)$		
$\mu_{v, \beta}$		
N_i		
$N_s(y)$		