

**Seleção de modelos para segmentação
de sequências simbólicas usando
máxima verossimilhança penalizada**

Bruno Monte de Castro

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

Área de Concentração: **Estatística**

Orientador: **Prof. Dr^a. Florencia Graciela Leonardi**

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro do CNPq

- São Paulo, fevereiro de 2013 -

**Seleção de modelos para segmentação
de sequências simbólicas usando
máxima verossimilhança penalizada**

Esta versão definitiva da dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa realizada por Bruno Monte de Castro em 20/02/2013.

Comissão Julgadora:

- Prof^a. Dr^a Florencia Graciela Leonardi (orientadora) - IME-USP.
- Prof. Dr. Anatoli Iambartsev - IME-USP.
- Prof. Dr. Jesús Enrique Garcia - UNICAMP.

“In fearful day, in raging night,
With strong hearts full, our souls ignite,
When all seems lost in the War of Light,
Look to the stars - For hope burns bright!”

Saint Walker

Agradecimentos

Gostaria de agradecer:

Aos meus pais Ademar e Arlene pelo apoio, carinho, educação e por estarem sempre ao meu lado. Aos meus irmãos/irmãs Jamilys, Janaira, Tiago e Priscila (que é minha sobrinha, mas praticamente uma irmã), pela força e principalmente por acreditarem em mim. As minhas tias Amélia e Célia que sempre me apoiaram e me ajudaram bastante. A Mônica por ter me ajudado muito nesse período.

A minha orientadora Prof^a. Florencia Graciela Leonardi pela paciência, respeito e ensinamentos. Muito obrigado pela ótima orientação que recebi neste trabalho.

Aos professores e funcionários do IME-USP, em especial aos Prof. Luís Gustavo Esteves e Prof. Fábio Prates Machado.

Aos professores e funcionários da Universidade Federal do Ceará, especialmente a Prof^a. Silvia Maria de Freitas, Juvêncio Santos Nobre, José Aílton Alencar Andrade. Queria fazer um agradecimento especial ao Prof. João Maurício que acreditou em mim desde o começo e sem os seus conselhos, ensinamentos, incentivo eu não estaria no mestrado, MUITO OBRIGADO!!!

Aos meus colegas da UFC e do IME, em especial aos meus amigos da turma 2007.1, que sempre acreditaram na minha pessoa e pelos estudos em grupo. Obrigado também Rafael Farias se não fosse seu incentivo no Sinape de 2010 para vir a SP e fazer a prova de admissão eu não estaria hoje onde estou.

Resumo

Castro, B. M. **Seleção de modelos para segmentação de sequências simbólicas usando máxima verossimilhança penalizada.** 66 f. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2013.

O problema de segmentação de sequências tem o objetivo de particionar uma sequência ou um conjunto delas em um número finito de segmentos distintos tão homogêneos quanto possível. Neste trabalho consideramos o problema de segmentação de um conjunto de sequências aleatórias, com valores em um alfabeto \mathcal{A} finito, em um número finito de blocos independentes. Supomos ainda que temos m sequências independentes de tamanho n , construídas pela concatenação de s segmentos de comprimento l_j^* , sendo que cada bloco é obtido a partir da distribuição \mathbb{P}_j em $\mathcal{A}^{l_j^*}$, $j = 1, \dots, s$. Além disso denotamos os verdadeiros pontos de corte pelo vetor $\mathbf{k}^* = (k_1^*, \dots, k_{s-1}^*)$, com $k_i^* = \sum_{j=1}^i l_j^*$, $i = 1, \dots, s-1$, esses pontos representam a mudança de segmento. Propomos usar o critério da máxima verossimilhança penalizada para inferir simultaneamente o número de pontos de corte e a posição de cada um desses pontos. Também apresentamos um algoritmo para segmentação de sequências e realizamos algumas simulações para mostrar seu funcionamento e sua velocidade de convergência. Nosso principal resultado é a demonstração da consistência forte do estimador dos pontos de corte quando o m tende ao infinito.

Palavras-chave: Segmentação de sequências, máxima verossimilhança penalizada, consistência forte.

Abstract

Castro, B. M. **A model selection criterion for the segmentation of symbolic sequences using penalized maximum likelihood.** 66 f. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2013.

The sequence segmentation problem aims to partition a sequence or a set of sequences into a finite number of segments as homogeneous as possible. In this work we consider the problem of segmenting a set of random sequences with values in a finite alphabet \mathcal{A} into a finite number of independent blocks. We suppose also that we have m independent sequences of length n , constructed by the concatenation of s segments of length l_j^* and each block is obtained from the distribution \mathbb{P}_j over $\mathcal{A}^{l_j^*}$, $j = 1, \dots, s$. Besides we denote the real cut points by the vector $\mathbf{k}^* = (k_1^*, \dots, k_{s-1}^*)$, with $k_i^* = \sum_{j=1}^i l_j^*$, $i = 1, \dots, s-1$, these points represent the change of segment. We propose to use a penalized maximum likelihood criterion to infer simultaneously the number of cut points and the position of each one those points. We also present a algorithm to sequence segmentation and we present some simulations to show how it works and its convergence speed. Our principal result is the proof of strong consistency of this estimators when m grows to infinity.

Keywords: Sequence segmentation, penalized maximum likelihood, strong consistency.

Conteúdo

1	Introdução	11
1.1	O Modelo Probabilístico	12
1.2	Objetivos	13
1.3	Organização do Trabalho	13
2	Inferência Estatística	15
2.1	Função de Verossimilhança	15
2.2	Estimador de Máxima Verossimilhança Penalizada	18
2.3	Resultados de Consistência	20
3	Demonstração da Consistência do Estimador	22
3.1	Demonstração do Lema 2.3.1	22
3.1.1	Caso em que $k_h^* < k_{h+1}^{(r)}$	24
3.1.2	Caso em que $k_{h+w}^{(r)} < k_h^* < k_{h+w+1}^{(r)}$	31
3.1.3	Caso em que $\mathbf{k}^{(r)} = (k_1^*, \dots, k_{s-1}^*, k_s^{(r)}, \dots, k_{t-1}^{(r)})$	35
3.1.4	Caso em que $\mathbf{k}^{(r)} = (k_1^*, \dots, k_{t-1}^*)$	38
3.2	Demonstração do Teorema 2.3.2	43
4	Simulações	44
4.1	Algoritmo de Estimação	44
4.2	Exemplos	44
A	Definições e Resultados Básicos	49
B	Rotinas em R	53
	Referências Bibliográficas	65

Lista de Tabelas

2.1	<i>Probabilidade empírica de cada sequência na amostra do Exemplo 2.1.1.</i>	17
4.1	<i>Logaritmo da função de verossimilhança penalizada para diferentes número de segmentos no Exemplo 4.2.1.</i>	45
4.2	<i>Logaritmo da função de verossimilhança penalizada para diferentes número de segmentos no Exemplo 4.2.2.</i>	46

Lista de Figuras

1.1	<i>Representação de 2 segmentos em uma dada sequência de tamanho n.</i>	13
2.1	<i>Representação de t segmentos em m sequências de tamanho n.</i>	16
2.2	<i>Árvore de Contexto obtida através do segmento aleatório X_2.</i>	19
2.3	<i>Gráfico comparativo do logaritmo da função de verossimilhança com e sem a penalização em relação ao número de segmentos estimados.</i>	19
2.4	<i>Gráfico do logaritmo da função de verossimilhança com e sem a penalização em relação ao ponto de corte k_1, no modelo com 2 segmentos.</i>	20
3.1	<i>Ilustração dos casos b, a e c nas iterações $r + 1$, $r + 2$, $r + 3$, respectivamente.</i>	23
3.2	<i>Ilustração do caso d.</i>	24
3.3	<i>Possíveis atualizações quando $k_h^* < k_{h+1}^{(r)}$</i>	25
3.4	<i>Ilustração do vetor $\mathbf{k}^{(r)}$ em (3.2).</i>	25
3.5	<i>Ilustração do vetor $\mathbf{k}^{(r+1)}$ em (3.2) e quando $k_h^* < k_{h+1}^{(r)}$.</i>	26
3.6	<i>Atualização da iteração r para $r + 1$, quando $k_h^* < k_{h+1}^{(r)}$.</i>	26
3.7	<i>Ilustração dos vetores $\mathbf{k}^{(r)}$ e $\mathbf{k}^{(r+1)}$ em (3.2) quando $k_{h+w}^{(r)} < k_h^* < k_{h+w+1}^{(r)}$.</i>	31
3.8	<i>Atualização da iteração r para $r + 1$, quando $k_{h+w}^{(r)} < k_h^* < k_{h+w+1}^{(r)}$.</i>	32
3.9	<i>Esboço de um possível vetor $\mathbf{k}^{(r)} = (k_1^*, \dots, k_{s-1}^*, k_s^{(r)}, \dots, k_{t-1}^{(r)})$.</i>	35
3.10	<i>Ilustração do vetor $\mathbf{k}^{(r+1)}$ em (3.17).</i>	37
3.11	<i>Remoção do ponto $k_s^{(r)}$ na iteração r para $r + 1$.</i>	37
3.12	<i>Ilustração do vetor $\mathbf{k}^{(r)} = (k_1^*, \dots, k_{t-1}^*)$.</i>	39
3.13	<i>Ilustração do vetor $\mathbf{k}^{(r+1)}$ em (3.20)</i>	39
3.14	<i>Adição do ponto k_t^* da iteração r para $r + 1$.</i>	41
4.1	<i>Gráfico do logaritmo da função de verossimilhança penalizada em relação ao ponto de corte k_1 quando existem 2 segmentos no Exemplo 4.2.1.</i>	45
4.2	<i>Árvore de Contexto obtida a partir do segmento aleatório X_3.</i>	46

4.3	<i>Gráfico do logaritmo da função de verossimilhança penalizada em relação ao ponto de corte nos casos com 2 e 3 segmentos, respectivamente.</i>	47
4.4	<i>Gráfico do logaritmo da função de verossimilhança penalizada em relação ao ponto de corte em 2 segmentos, para diferentes valores de m.</i>	47
4.5	<i>Gráfico do logaritmo da função de verossimilhança penalizada em relação ao ponto de corte em 2 segmentos quando $m = 25000$.</i>	48

Capítulo 1

Introdução

O problema de segmentação de sequências têm o objetivo de particionar uma sequência ou um conjunto delas em um número finito de segmentos distintos tão homogêneos quanto possível. Este problema pode ser aplicado em diversas áreas tais como a bioinformática, previsão do tempo e telecomunicações. Na bioinformática o maior problema é segmentar um gene ou uma proteína em diferentes partes funcionais.

Nos últimos anos foram apresentados vários métodos para segmentar sequências, principalmente aplicados a segmentação do DNA. O DNA, sigla em inglês para ácido desoxirribonucléico, é o material genético das células que pode ser interpretado como um código contendo instruções para as funções do corpo. O DNA é composto de quatro bases: Adenina (A), Citosina (C), Guanina (G) e Timina (T). Uma sequência das bases do DNA transmite informações para que as células fabriquem uma certa proteína específica, esse fato pode ser comparado a construção de uma frase, na qual devemos alocar as letras para que uma sentença faça sentido.

Braun & Muller (1998) apresentam várias técnicas estatísticas para auxiliar no processo de segmentação de sequências, dentre essas técnicas destacamos a segmentação usando cadeias de Markov ocultas e o método de mudança-de-ponto (*change-point*). Boys *et al* (2000) ilustram através de um exemplo, como realizar segmentações de sequências de DNA usando uma abordagem bayesiana para uma generalização das cadeias de Markov ocultas, já que no modelo apresentado o processo observado Y_t depende tanto de Y_{t-1} como do estado oculto S_t , quando o número de estados escondidos é conhecido e cada estado escondido corresponde a um tipo de segmento. Em Boys & Henderson (2004) estes resultados foram generalizados para o caso em que o número de segmentos é desconhecido e o processo observado segue uma cadeia de Markov de ordem q , isto é, o valor observado Y_t depende de Y_{t-1}, \dots, Y_{t-q} e S_t . Boys *et al* (2000) e Boys & Henderson (2004) utilizaram como priori para a transição das bases de DNA e para os tipos de segmento a distribuição Dirichlet com o intuito de obter prioris conjugadas, mas Nur *et al* (2009) perceberam, através de uma análise de sensibilidade, que uma mistura da distribuição Dirichlet como priori para os tipos de segmento proporcionava melhores estimativas.

Um dos primeiros algoritmos para segmentação de sequências, conhecido como Dividir-e-Conquistar, têm o objetivo de dividir uma dada sequência em segmentos homogêneos e realizar uma otimização local a partir da divergência de Jensen-Shannon, ver Bernaola-Galván *et al* (1996) e Bernaola-Galván *et al* (1999). Li (2001a) e Li (2001b) notou que esse método não possuía um critério de parada satisfatório e propôs que a segmentação fosse realizada usando ferramentas de seleção de modelos e adotando o BIC (Critério de Informação Bayesiana) como o critério de parada. Terzi & Tsaparas (2006) apresentaram o algoritmo conhecido como Dividir e Segmentar (DNS), em que a sequência é particionada em subsequências disjuntas (Dividir). Essas subsequências são segmentadas utilizando programação dinâmica em um número definido, k , de pontos de corte (Segmentar). Após essa segmentação novas sequências são formadas a partir da concatenação dos segmentos. O critério para essa concatenação é fazer uma ponderação pelo comprimento dos segmentos.

O artigo que motivou esse trabalho foi Gwadera *et al* (2008) que propuseram um algoritmo para determinar o número ótimo de segmentos que particiona a sequência e ajusta uma cadeia de Markov de alcance variável ideal para cada segmento. Para selecionar o número de segmentos da sequência foram usados o Critério de Informação Bayesiana (BIC) e uma modificação do princípio de Mínimo Comprimento de Descrição (MDL) que usa o estimador Krichevsky-Trofimov (KT) (Csiszár & Talata (2006)). Para resolver o problema do número ótimo de segmentos foi apresentado o algoritmo chamado *TreeSegment*, que se baseia nos algoritmos de Contexto (ver Rissanen (1983)) e de Maximização da Árvore de Contexto (Willems *et al* (2000)), que tem a função de podar os nós terminais no BIC e no MDL, respectivamente. Gwadera *et al* (2008) observaram que o modelo é afetado pelas bordas dos segmentos e inseriram uma penalidade para resolver esse problema, que pode ser entendida em termos da probabilidade de transição entre os estados ocultos (que correspondem aos segmentos) na cadeia de Markov oculta.

1.1 O Modelo Probabilístico

Neste trabalho vamos considerar o problema de segmentação de um conjunto de sequências simbólicas aleatórias com valores em uma alfabeto finito com cardinalidade $|\mathcal{A}|$. Ao longo do texto vamos apresentar alguns termos e notações para um melhor entendimento do problema. Denotamos por $a_u^v = a_u a_{u+1} \cdots a_v$, $1 \leq u < v$, uma sequência de tamanho $v - u + 1$ e cada componente $a_t \in \mathcal{A}$. Chamamos de “palavra” uma configuração qualquer de uma subsequência de a_u^v . Na literatura é muito comum concatenar duas ou mais palavras criando uma nova sequência. Para denotar essa operação supomos duas sequências, $b = b_u^v \in \mathcal{A}^{v-u+1}$ e $c = c_{v+1}^w \in \mathcal{A}^{w-v}$, assim dizemos que bc é a concatenação de b e c em \mathcal{A}^{w-u+1} .

Seja Y uma sequência aleatória sobre \mathcal{A}^n , tal que Y é a concatenação de s segmentos aleatórios independentes X_1, X_2, \dots, X_s , sendo que cada segmento X_j assume uma distribuição de proba-

bilidade \mathbb{P}_j , $j = 1, \dots, s$. Assim, temos que $X_j \in \mathcal{A}^{l_j^*}$, com $\mathcal{A}^{l_j^*} = \{(a_1, \dots, a_{l_j^*}) : a_t \in \mathcal{A}\}$ representando as possíveis palavras no segmento de tamanho l_j^* e conseqüentemente $\sum_{j=1}^s l_j^* = n$. Portanto, temos uma seqüência de valores discretos de tamanho n dividida em s blocos independentes, de tamanho finito, sendo que cada segmento tem comprimento l_j^* .

Denotamos pelo vetor $\mathbf{k}^* = (k_1^*, \dots, k_{s-1}^*)$ os verdadeiros pontos de corte (independência) entre os blocos. Esses pontos de independência podem ser obtidos em função do tamanho dos segmentos, da seguinte forma

$$\begin{aligned} k_1^* &= l_1^* \\ k_2^* &= l_1^* + l_2^* \\ &\vdots \\ &\vdots \\ k_{s-1}^* &= \sum_{j=1}^{s-1} l_j^*. \end{aligned}$$

Na Figura 1.1 mostramos uma seqüência em que $n = 10$ e $s=2$ e temos a distribuição de X_1 sobre $\mathcal{A}^{l_1^*}$ e X_2 em $\mathcal{A}^{l_2^*}$, com $l_1^*=4$ e $l_2^* = 6$.

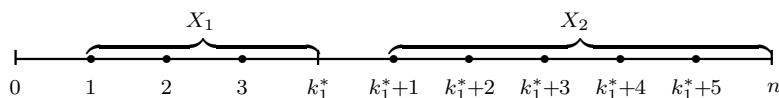


Figura 1.1: Representação de 2 segmentos em uma dada seqüência de tamanho n .

1.2 Objetivos

O objetivo deste trabalho é, a partir de uma amostra de m seqüências, obter um estimador para os pontos de corte usando, como critério de informação para a seleção do modelo, o método da máxima verossimilhança penalizada. Além disso, vamos apresentar um algoritmo de segmentação e estudá-lo através de simulações. Nosso principal resultado é a demonstração da consistência forte desse estimador, quando o número de seqüências tende ao infinito.

1.3 Organização do Trabalho

No Capítulo 2 definimos a função de verossimilhança do modelo e argumentamos sobre a necessidade de uma penalização para a mesma. Além disso, enunciemos um Lema e um Teorema que garantem a consistência forte do estimador. No Capítulo 3 demonstramos todos os possíveis casos

do Lema e a partir dele demostramos o Teorema. No Capítulo 4 apresentamos o algoritmo de segmentação e alguns exemplos com o intuito de validar o seu funcionamento. O Apêndice A é reservado para algumas definições e resultados básicos necessários na demonstração do Lema no Capítulo 3. Por fim, o Apêndice B traz o script em R dos exemplos no Capítulo 4.

Capítulo 2

Inferência Estatística

Neste capítulo apresentamos a função de verossimilhança e através de um exemplo vemos a necessidade de inserir uma penalização, com o intuito de obter o número ideal de pontos de corte. Com isso definimos o estimador dos pontos de corte baseado na função verossimilhança penalizada. Por último, a fim de provar a consistência forte do estimador apresentamos o Lema 2.3.1 e o Teorema 2.3.2.

2.1 Função de Verossimilhança

Com o intuito de realizar inferência para nosso modelo supomos que temos m sequências amostrais de tamanho n , cada uma delas independentes entre si. Em nosso problema de segmentação um dado segmento qualquer é referente às m sequências, isto é, o j -ésimo bloco é obtido pelo estudo de toda a amostra. Seja $\mathbf{k} = (k_1, \dots, k_{t-1})$ um vetor qualquer de pontos de corte, de forma que t pode ser maior, menor ou igual a s , além disso denotamos, para simplificar a notação, o ponto 0 como k_0 e n -ésimo ponto como k_t , ver Figura 2.1. Inicialmente vamos considerar somente o estudo do segmento j de comprimento $l_j = k_j - k_{j-1}$, que é o bloco limitado pelos pontos $k_{j-1} + 1$ e k_j . Representamos por $y_{k_{j-1}+1}^{k_j} = y_{k_{j-1}+1} y_{k_{j-1}+2} \dots y_{k_j}$, qualquer sequência possível dos símbolos em \mathcal{A}^{l_j} , e denotamos por $N_m^{\{k_{j-1}:k_j\}} \left(a_{k_{j-1}+1}^{k_j} \right)$ o número de vezes que aparece a palavra $a_{k_{j-1}+1}^{k_j}$ no j -ésimo segmento nas m sequências. Essa variável pode ser definida da seguinte forma

$$N_m^{\{k_{j-1}:k_j\}} \left(a_{k_{j-1}+1}^{k_j} \right) = \sum_{i=1}^m \mathbb{1} \left(y_j^{(i)} = a_{k_{j-1}+1}^{k_j} \right)$$

em que $\mathbb{1}$ é a função indicadora e $y_j^{(i)}$ representa a palavra no segmento j correspondente à i -ésima observação na amostra.

O estimador de máxima verossimilhança de $\mathbb{P}\left(Y_{k_{j-1}+1}^{k_j} = a_{k_{j-1}+1}^{k_j}\right)$ é dado por

$$\hat{\mathbb{P}}\left(Y_{k_{j-1}+1}^{k_j} = a_{k_{j-1}+1}^{k_j}\right) = \hat{\mathbb{P}}_{\{k_{j-1}:k_j\}}\left(a_{k_{j-1}+1}^{k_j}\right) = \frac{N_m^{\{k_{j-1}:k_j\}}\left(a_{k_{j-1}+1}^{k_j}\right)}{m}.$$

Esse resultado pode ser demonstrado considerando uma amostra de tamanho m de distribuições Bernoulli, sendo sucesso aparecer a palavra $a_{k_{j-1}+1}^{k_j}$ na i -ésima sequência. Assim a função de verossimilhança empírica, em relação ao j -ésimo segmento, é dada por

$$\prod_{i=1}^m \hat{\mathbb{P}}\left(Y_{k_{j-1}+1}^{k_j} = y_j^{(i)}\right) = \prod_{a_{k_{j-1}+1}^{k_j} \in \mathcal{A}^{l_j}} \left(\frac{N_m^{\{k_{j-1}:k_j\}}\left(a_{k_{j-1}+1}^{k_j}\right)}{m} \right)^{N_m^{\{k_{j-1}:k_j\}}\left(a_{k_{j-1}+1}^{k_j}\right)}.$$

Adotamos, por conveniência, que $0^0 = 1$, isso ocorre quando uma dada palavra não aparece na amostra.

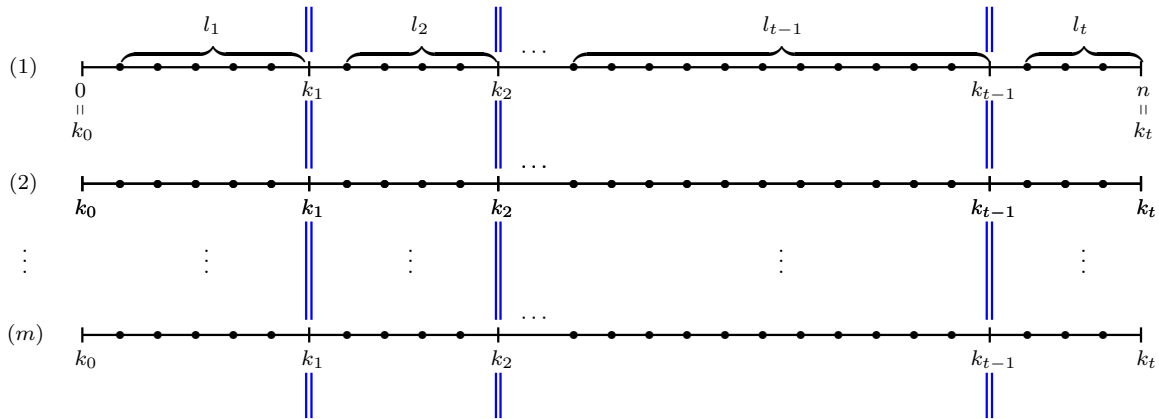


Figura 2.1: Representação de t segmentos em m sequências de tamanho n .

Exemplo 2.1.1. Considere um único bloco, denotado por j , e vamos calcular a probabilidade empírica de aparecer a sequência $a_{k_{j-1}+1}^{k_j}$ em uma amostra de $m = 12$ sequências com $l_j = 5$ e alfabeto $\mathcal{A} = \{b, c\}$. A partir dessa amostra obtemos as probabilidades empíricas para cada palavra, ver Tabela 2.1.

c c c b c
 c b c c c
 b c c b b
 c b c c c
 c b c b b
 c b b c b
 b c b b b
 c c c b c
 b c c c b
 c b c c c
 b c b c c
 c b b c b

Palavra	$\hat{\mathbb{P}}_{\{k_{j-1}:k_j\}} \left(a_{k_{j-1}+1}^{k_j} \right)$
c c c b c	1/6
c b c c c	1/4
b c c b b	1/12
c b c b b	1/12
c b b c b	1/6
b c b b b	1/12
b c c c b	1/12
b c b c c	1/12

Tabela 2.1: Probabilidade empírica de cada sequência na amostra do Exemplo 2.1.1.

A probabilidade empírica conjunta de uma i -ésima sequência pode ser escrita da forma

$$\hat{\mathbb{P}} \left(Y_{k_0+1}^{k_1} = y_1^{(i)}, Y_{k_1+1}^{k_2} = y_2^{(i)}, \dots, Y_{k_{t-1}+1}^{k_t} = y_t^{(i)} \right)$$

assim, a função de verossimilhança empírica do vetor \mathbf{k} correspondente aos dados $\mathbf{Y}_{m \times 1}$ observada é dada por

$$L(\mathbf{k}|\mathbf{Y}) = \prod_{i=1}^m \prod_{j=1}^t \hat{\mathbb{P}} \left(Y_{k_{j-1}+1}^{k_j} = a_{k_{j-1}+1}^{k_j} \right),$$

pois assumimos que cada bloco é independente. Tomando o logaritmo da função de verossimilhança empírica de \mathbf{k} dado \mathbf{Y} obtemos

$$l(\mathbf{k}|\mathbf{Y}) = \sum_{j=1}^t \sum_{a_{k_{j-1}+1}^{k_j}} N_m^{\{k_{j-1}:k_j\}} \left(a_{k_{j-1}+1}^{k_j} \right) \log \left(\frac{N_m^{\{k_{j-1}:k_j\}} \left(a_{k_{j-1}+1}^{k_j} \right)}{m} \right).$$

2.2 Estimador de Máxima Verossimilhança Penalizada

Quando estimamos os pontos de corte, a partir da função de verossimilhança percebemos que quanto menor o número de pontos estimados, ou seja, quanto menor for o número de segmentos maior será o valor da verossimilhança do modelo. Para prevenir que sejam estimados mais (ou menos) pontos de corte do que o necessário, estimamos o vetor \mathbf{k} usando o método da máxima verossimilhança penalizada. Com isso, o logaritmo da função de verossimilhança penalizada é dado por

$$LVP(\mathbf{k}|\mathbf{Y}) = l(\mathbf{k}|\mathbf{Y}) - c \left(\sum_{j=1}^t |\mathcal{A}|^{l_j} - t \right) \log m \quad (2.1)$$

sendo c uma constante positiva e a penalização baseada tanto no número de segmentos estimados quanto no tamanho de cada um desses segmento.

Definição 2.2.1. *Dada uma constante $c > 0$, o estimador de \mathbf{k}^* , em relação a um tamanho de amostra m é dado por*

$$\hat{\mathbf{k}}_m = \underset{\substack{\mathbf{k}: k_1 < k_2 < \dots < k_{t-1} \\ 1 \leq t \leq n}}{\text{argmax}} \left\{ l(\mathbf{k}|\mathbf{Y}) - c \left(\sum_{j=1}^t |\mathcal{A}|^{l_j} - t \right) \log m \right\}.$$

Na prática, esse estimador é bem difícil de ser calculado quando o tamanho da sequência é grande e/ou para quando existem muitos segmentos. Com a intenção de resolver esse problema, será apresentado no Capítulo 4 um algoritmo iterativo que se utiliza de recorrências e permitirá encontrar os estimadores dos pontos de corte de maneira mais viável.

Exemplo 2.2.2. Realizamos uma simulação com $m = 5000$ sequências amostrais de tamanho $n = 6$ obtidas a partir da sequência aleatória Y sobre o alfabeto $\mathcal{A} = \{1, 2\}$, sendo que Y é composta por dois segmentos independentes X_1 e X_2 e com o ponto de corte $k_1^* = 3$ de forma que

- X_1 é gerado a partir de uma cadeia de Markov com a seguinte matriz de transição

$$P_1 = \begin{matrix} & \begin{matrix} 1 & 2 \end{matrix} \\ \begin{matrix} 1 \\ 2 \end{matrix} & \begin{pmatrix} 0,3 & 0,7 \\ 0,8 & 0,2 \end{pmatrix} \end{matrix}$$

- O segmento X_2 é gerado através de uma cadeia de Markov de alcance variável de ordem 2 com 3 folhas, como na Figura 2.2, ver Bühlmann & Wyner (1999) e Rissanen (1983). Foi utilizado o pacote VLMC do software R para essa simulação, mais detalhes podem ser encontrados em Mächler & Bühlmann (2002).

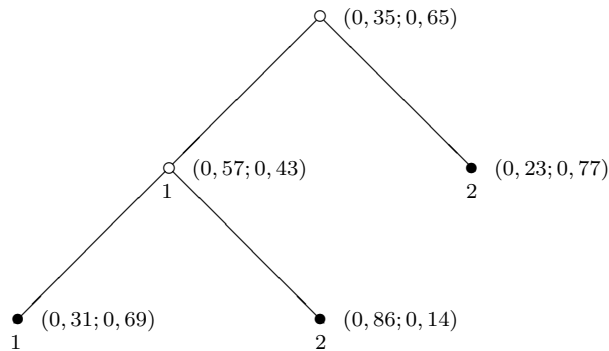


Figura 2.2: *Árvore de Contexto obtida através do segmento aleatório X_2 .*

Primeiro devemos encontrar quantos pontos de corte existem no modelo. Quando calculamos a função de verossimilhança percebemos que ela é decrescente em relação ao número de pontos de corte/segmentos estimados e seu valor será máximo quando nenhum ponto de corte é estimado. Esse resultado faz sentido, já que quanto maior for o segmento, menor será a chance de uma dada palavra se repetir. Por conta disso, propomos a função de verossimilhança penalizada dada em (2.1) que a retorna como valor de máximo o número correto de pontos de corte, como mostrado na Figura 2.3.

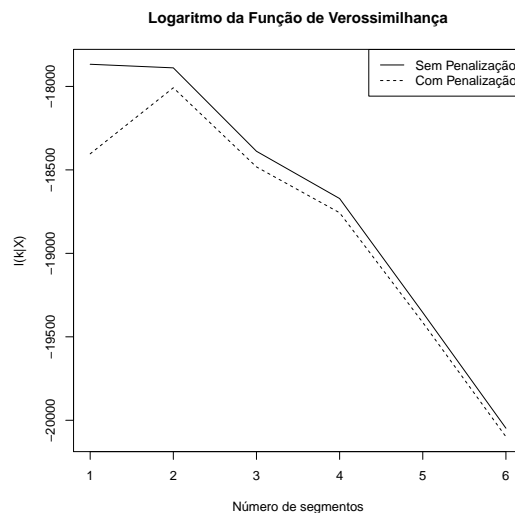


Figura 2.3: *Gráfico comparativo do logaritmo da função de verossimilhança com e sem a penalização em relação ao número de segmentos estimados.*

Agora que conhecemos o número ideal de segmentos do modelo temos que saber qual o ponto

de corte que maximiza a função de verossimilhança. A Figura 2.4 mostra que o ponto de corte $k_1 = 3$ é o ponto de máximo tanto na função de verossimilhança quanto na função de verossimilhança penalizada. Neste exemplo, tanto a função de verossimilhança com penalização como a sem penalização parecem ter o mesmo comportamento, isso se deve porque n é pequeno e a penalização não é afetada tão drasticamente.

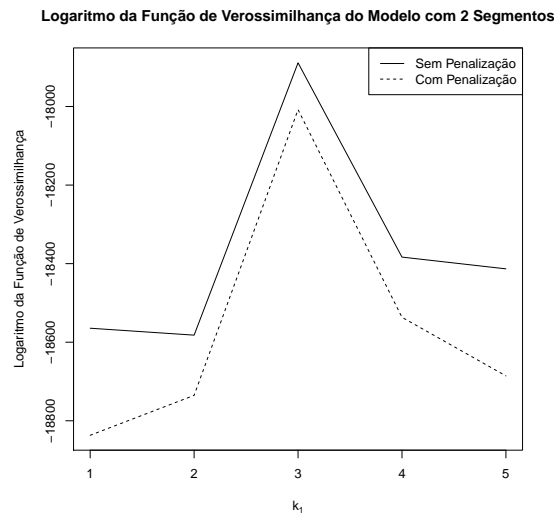


Figura 2.4: Gráfico do logaritmo da função de verossimilhança com e sem a penalização em relação ao ponto de corte k_1 , no modelo com 2 segmentos.

2.3 Resultados de Consistência

Lema 2.3.1. *Seja $\mathbf{k}^{(r)} \neq \mathbf{k}^*$. Existe um vetor $\mathbf{k}^{(r+1)} \neq \mathbf{k}^{(r)}$ tal que*

$$\frac{LVP(\mathbf{k}^{(r)}|\mathbf{Y}) - LVP(\mathbf{k}^{(r+1)}|\mathbf{Y})}{m} \leq c(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) < 0,$$

quase certamente, quando $m \rightarrow \infty$ e $c(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)})$ é uma constante que depende dos pontos de corte estimados na iteração r e $r + 1$.

O Lema 2.3.1 diz que, dado um vetor qualquer de segmentos estimados existe um outro vetor com melhores estimações, de forma que $LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}) > LVP(\mathbf{k}^{(r)}|\mathbf{Y})$, quando o tamanho da amostra é grande.

O vetor $\mathbf{k}^{(r+1)}$ é uma atualização de $\mathbf{k}^{(r)}$, sendo que o vetor $\mathbf{k}^{(r)}$ pode apresentar pelo menos um ponto de corte estimado erroneamente ou ter mais ou menos pontos de corte estimados em relação ao vetor \mathbf{k}^* .

Em cada modificação atualizamos o valor de $k_j^{(r)}$ para o verdadeiro ponto de corte k_j^* , não

importando o tipo de erro que o modelo possui. Nesta atualização verificamos para qual valor de $k_j^{(r)}$ a verossimilhança penalizada é máxima.

Teorema 2.3.2. *O estimador dado na Definição 2.2.1 satisfaz*

$$\hat{\mathbf{k}}_m = \mathbf{k}^*$$

para m suficientemente grande com probabilidade 1.

O Teorema 2.3.2 garante a consistência forte do estimador do vetor \mathbf{k}^* , em um número finito de iterações, quando o número de sequências tende ao infinito. Podemos dizer que quanto menos pontos de corte estimados erroneamente mais rápida será a convergência.

Capítulo 3

Demonstração da Consistência do Estimador

No presente capítulo vamos demonstrar a consistência do estimador dos pontos de corte. Primeiro, iremos realizar a demonstração do Lema 2.3.1, que está dividida nas subseções 3.1.1, 3.1.2, 3.1.3 e 3.1.4. Logo após, demonstramos o Teorema 2.3.2 usando o Lema 2.3.1 como principal argumento.

3.1 Demonstração do Lema 2.3.1

Considere um vetor de pontos de corte estimados denotado por $\mathbf{k}^{(r)}$, em que $\mathbf{k}^{(r)} \neq \mathbf{k}^*$. Vamos atualizar as componentes $k_j^{(r)}$ que são diferentes de k_j^* na situação em que todos os pontos de corte estimados antes de $k_j^{(r)}$ sejam iguais aos verdadeiros pontos de corte. Além disso, essas atualizações deverão respeitar a seguinte restrição

$$k_1^{(r)} < k_2^{(r)} < \dots < k_{t-1}^{(r)}. \quad (3.1)$$

Suponha agora que h é o primeiro ponto de $\mathbf{k}^{(r)}$ diferente de \mathbf{k}^* , isto é

$$\mathbf{k}^{(r)} = \left(k_1^*, k_2^*, \dots, k_{h-1}^*, k_h^{(r)}, \dots, k_{t-1}^{(r)} \right), \text{ com } k_h^{(r)} \neq k_h^*. \quad (3.2)$$

é importante ressaltar que os pontos $k_{h+1}^{(r)}, \dots, k_{t-1}^{(r)}$ podem ser ou não os verdadeiros pontos de corte. Dessa forma, na atualização do ponto $k_h^{(r)}$ para k_h^* devemos considerar os casos em que existe h tal que (3.2) satisfaz:

- a) $k_h^* < k_{h+1}^{(r)}$;
- b) $k_{h+w}^{(r)} < k_h^* < k_{h+w+1}^{(r)}$, para algum $w \geq 1$.

uma atualização, nesse contexto, é uma mudança de posição do ponto de corte estimado quando o mesmo não está na sua posição verdadeira. Em contrapartida, há situações em que não existe

h que satisfaz o vetor em (3.2). Isso acontece quando necessitamos adicionar ou remover alguns pontos de corte, assim a configuração do vetor $\mathbf{k}^{(r)}$ pode ser:

c) $\mathbf{k}^{(r)} = (k_1^*, \dots, k_{s-1}^*, k_s^{(r)}, \dots, k_{t-1}^{(r)})$, com $t > s$;

d) $\mathbf{k}^{(r)} = (k_1^*, \dots, k_{t-1}^*)$, com $t < s$.

No caso **a** o ponto de corte k_h^* está localizado antes do ponto $k_{h+1}^{(r)}$ e que satisfaz a restrição em (3.1). Em **b** essa restrição não é respeitada, portanto além de atualizar o ponto k_h^* devemos ainda desprezar os pontos estimados de $k_{h+1}^{(r)}$ até $k_{h+w}^{(r)}$. Em **c** foram estimados mais pontos de corte do que o necessário, de forma que já conhecemos os verdadeiros $s - 1$ pontos de independência, então vamos remover os pontos de corte excedentes. Por último, o caso **d** ocorre quando estimamos corretamente os primeiros $t - 1$ pontos de corte, sendo $t < s$ e devemos adicionar os demais pontos de corte, um por vez a cada iteração, até obter a mesma configuração do vetor \mathbf{k}^* . Observamos que, se ocorre o caso **c**, não será possível acontecer o caso **d** e vice-versa.

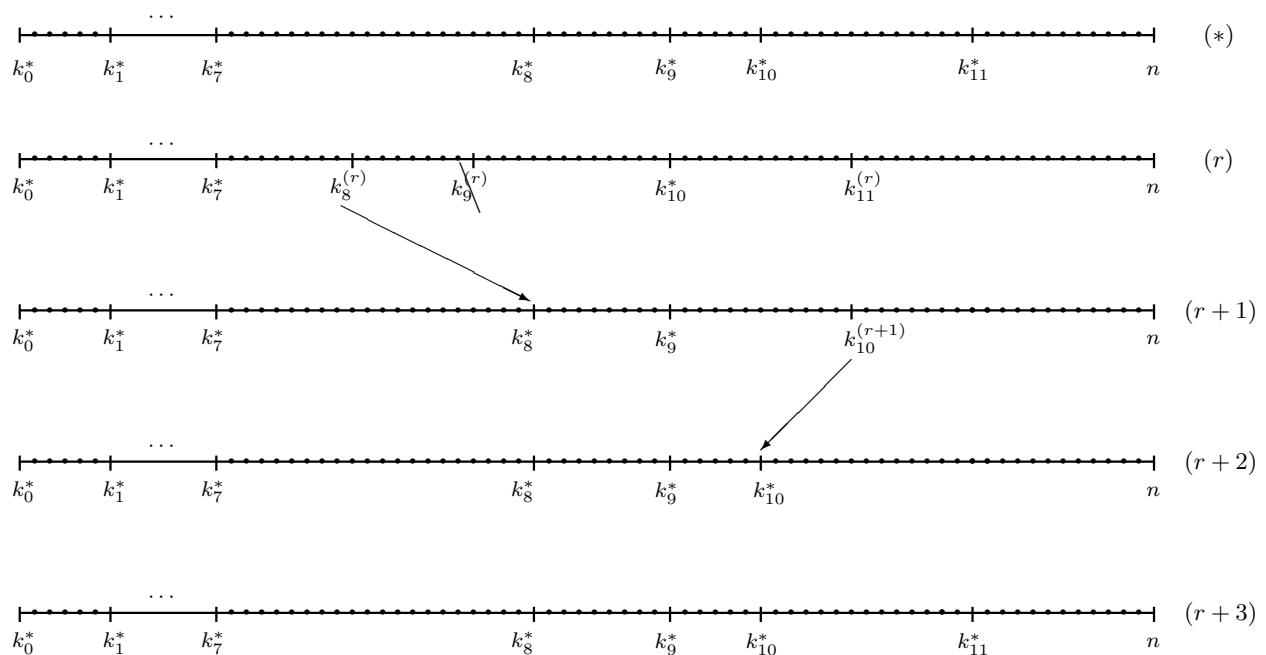


Figura 3.1: Ilustração dos casos **b**, **a** e **c** nas iterações $r + 1$, $r + 2$, $r + 3$, respectivamente.

Vamos supor um exemplo em que o número verdadeiro de pontos de corte é 11 e ilustramos o modelo correto em (*), como mostrado na Figura 3.1. Na iteração r o primeiro ponto que satisfaz

$k_j^{(r)} \neq k_j^*$ é o $k_8^{(r)}$. Vemos que a atualização do ponto $k_8^{(r)}$ satisfaz o caso **b** quando $w = 1$ e além disso, o ponto $k_9^{(r)}$ é desprezado fazendo com que todos os pontos posteriores mudem seus índices na iteração $r + 1$. Em seguida notamos que o caso **a** é aplicado na atualização do ponto $k_{10}^{(r+1)}$, assim no passo $r + 2$ não sobrou nenhum ponto estimado para atualizar. Como precisamos atualizar mais um ponto vamos usar o caso **c** e adicionar o último ponto de independência. Na Figura 3.2, considerando mesmo o exemplo do modelo com 11 pontos de corte verdadeiros, foram estimados mais pontos de corte do que o necessário. Assim, exibimos a remoção do ponto $k_{12}^{(r)}$ entre os passos r e $r + 1$ quando foram encontrados todos os pontos de independência do modelo. Para realizar a demonstração dos casos **a**, **b** e **c**, vamos considerar a seguinte diferença

$$\ell_m(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) = \frac{LVP(\mathbf{k}^{(r)}|\mathbf{Y}) - LVP(\mathbf{k}^{(r+1)}|\mathbf{Y})}{m}. \quad (3.3)$$

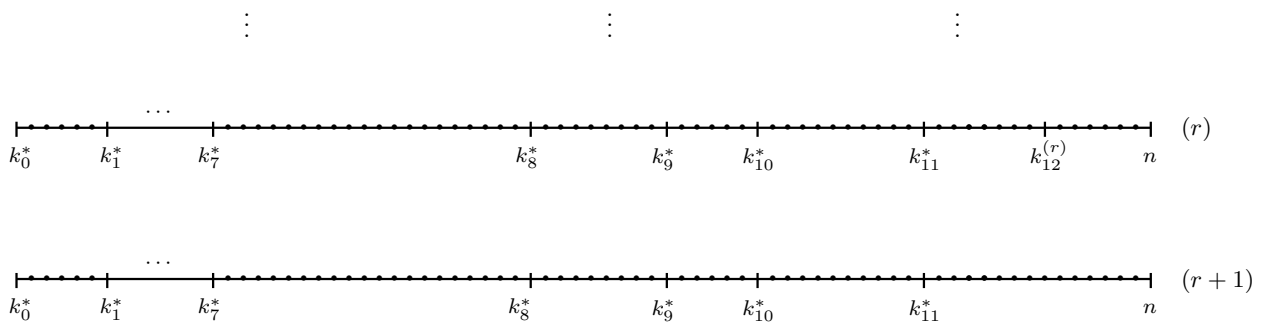


Figura 3.2: Ilustração do caso d.

3.1.1 Caso em que $k_h^* < k_{h+1}^{(r)}$

Nesta subseção consideramos o vetor em (3.2) e o caso em que $k_h^* < k_{h+1}^{(r)}$. É mostrado na Figura 3.3 (a) e (b) a atualização quando $k_h^* < k_h^{(r)}$ ou $k_h^* > k_h^{(r)}$, respectivamente.

Na demonstração iremos considerar a situação em que $k_h^* > k_h^{(r)}$, dessa forma as componentes do vetor $\mathbf{k}^{(r+1)}$, após a atualização, são dadas por

$$k_j^{(r+1)} = \begin{cases} k_j^*, & 1 \leq j \leq h \\ k_j^{(r)}, & h + 1 \leq j \leq t - 1 \end{cases}$$

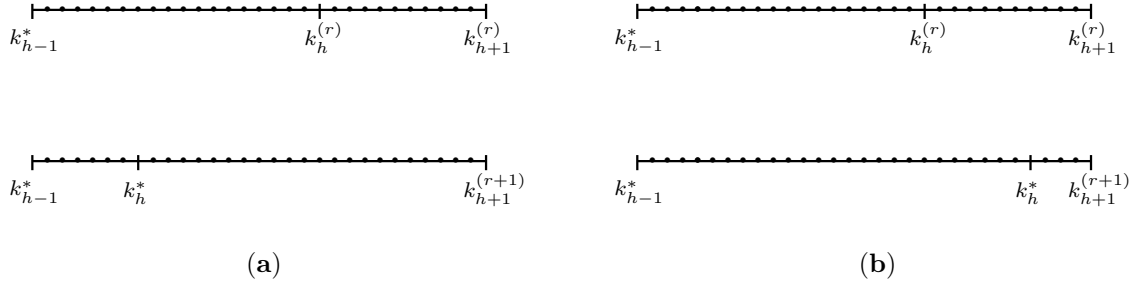


Figura 3.3: Possíveis atualizações quando $k_h^* < k_{h+1}^{(r)}$

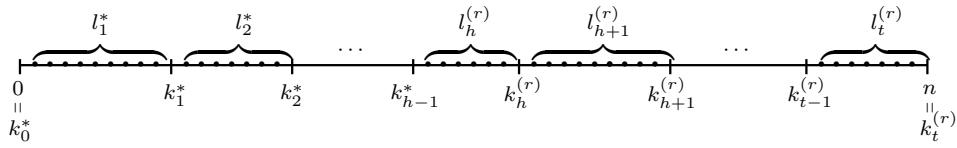


Figura 3.4: Ilustração do vetor $\mathbf{k}^{(r)}$ em (3.2).

Temos que o logaritmo da verossimilhança penalizada de $\mathbf{k}^{(r)}$ dado \mathbf{Y} (ver Figura 3.4) é da forma

$$\begin{aligned}
 LVP(\mathbf{k}^{(r)}|\mathbf{Y}) &= \sum_{j=1}^{h-1} \sum_{\substack{k_j^* \\ a_{k_{j-1}^*+1}^{k_j^*}}} N_m^{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right) \log \left(\frac{N_m^{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right)}{m} \right) \\
 &+ \sum_{\substack{k_h^{(r)} \\ a_{k_{h-1}^*+1}^{k_h^{(r)}}}} N_m^{\{k_{h-1}^*:k_h^{(r)}\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \log \left(\frac{N_m^{\{k_{h-1}^*:k_h^{(r)}\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right)}{m} \right) \\
 &+ \sum_{j=h+1}^t \sum_{\substack{k_j^{(r)} \\ a_{k_{j-1}^{(r)}+1}^{k_j^{(r)}}}} N_m^{\{k_{j-1}^{(r)}:k_j^{(r)}\}} \left(a_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \log \left(\frac{N_m^{\{k_{j-1}^{(r)}:k_j^{(r)}\}} \left(a_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right)}{m} \right) \\
 &- c \left(\sum_{j=1}^{h-1} |\mathcal{A}|^{l_j^*} + \sum_{j=h}^t |\mathcal{A}|^{l_j^{(r)}} - t \right) \log m. \tag{3.4}
 \end{aligned}$$

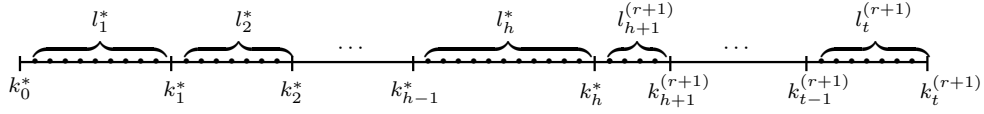


Figura 3.5: Ilustração do vetor $\mathbf{k}^{(r+1)}$ em (3.2) e quando $k_h^* < k_{h+1}^{(r)}$.

Por outro lado, no caso do vetor atualizado $\mathbf{k}^{(r+1)}$ temos

$$\begin{aligned}
LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}) &= \sum_{j=1}^h \sum_{\substack{k_j^* \\ a_{k_{j-1}^*+1}}} N_m^{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right) \log \left(\frac{N_m^{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right)}{m} \right) \\
&+ \sum_{\substack{k_{h+1}^{(r+1)} \\ a_{k_h^*+1}}} N_m^{\{k_h^*:k_{h+1}^{(r+1)}\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r+1)}} \right) \log \left(\frac{N_m^{\{k_h^*:k_{h+1}^{(r+1)}\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r+1)}} \right)}{m} \right) \\
&+ \sum_{j=h+2}^t \sum_{\substack{k_j^{(r+1)} \\ a_{k_{j-1}^{(r+1)}+1}}} N_m^{\{k_{j-1}^{(r+1)}:k_j^{(r+1)}\}} \left(a_{k_{j-1}^{(r+1)}+1}^{k_j^{(r+1)}} \right) \log \left(\frac{N_m^{\{k_{j-1}^{(r+1)}:k_j^{(r+1)}\}} \left(a_{k_{j-1}^{(r+1)}+1}^{k_j^{(r+1)}} \right)}{m} \right) \\
&- c \left(\sum_{j=1}^h |\mathcal{A}|^{l_j^*} + \sum_{j=h+1}^t |\mathcal{A}|^{l_j^{(r+1)}} - t \right) \log m
\end{aligned}$$

ver Figura 3.5.

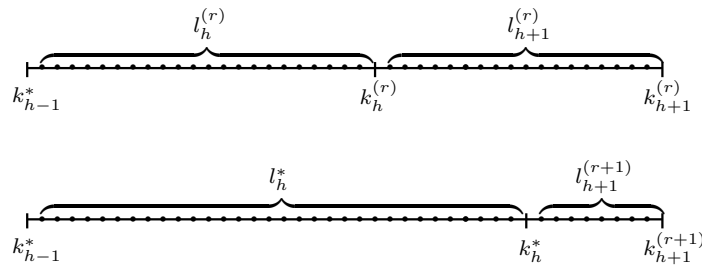


Figura 3.6: Atualização da iteração r para $r + 1$, quando $k_h^* < k_{h+1}^{(r)}$.

Dessa forma a expressão (3.3) se torna

$$\begin{aligned}
l_m(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) &= \sum_{\substack{k_h^{(r)} \\ a_{k_{h-1}^*+1}}} \hat{\mathbb{P}}_{\{k_{h-1}^*:k_h^{(r)}\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \log \hat{\mathbb{P}}_{\{k_{h-1}^*:k_h^{(r)}\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \\
&+ \sum_{\substack{k_{h+1}^{(r)} \\ a_{k_h^{(r)}+1}} \hat{\mathbb{P}}_{\{k_h^{(r)}:k_{h+1}^{(r)}\}} \left(a_{k_h^{(r)}+1}^{k_{h+1}^{(r)}} \right) \log \hat{\mathbb{P}}_{\{k_h^{(r)}:k_{h+1}^{(r)}\}} \left(a_{k_h^{(r)}+1}^{k_{h+1}^{(r)}} \right) \\
&- \sum_{\substack{k_h^* \\ a_{k_{h-1}^*+1}}} \hat{\mathbb{P}}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \log \hat{\mathbb{P}}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \\
&- \sum_{\substack{k_{h+1}^{(r+1)} \\ a_{k_h^*+1}}} \hat{\mathbb{P}}_{\{k_h^*:k_{h+1}^{(r+1)}\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r+1)}} \right) \log \hat{\mathbb{P}}_{\{k_h^*:k_{h+1}^{(r+1)}\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r+1)}} \right) \\
&- c \left(|\mathcal{A}|^{l_h^{(r)}} + |\mathcal{A}|^{l_{h+1}^{(r)}} - |\mathcal{A}|^{l_h^*} - |\mathcal{A}|^{l_{h+1}^{(r+1)}} \right) \frac{\log m}{m}. \tag{3.5}
\end{aligned}$$

como mostra a Figura 3.6. Para m grande, a probabilidade empírica, em um dado segmento, converge para a probabilidade teórica nos blocos independentes, da seguinte forma

$$\lim_{m \rightarrow \infty} \hat{\mathbb{P}}_{\{k_{j-1}^*:k_j^{(r)}\}} \left(a_{k_{j-1}^*+1}^{k_j^{(r)}} \right) = \begin{cases} \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^{(r)}} \right), & \text{se } k_j^* \geq k_j^{(r)} \\ \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right) \mathbb{P}_{\{k_j^*:k_v^*\}} \left(a_{k_j^*+1}^{k_j^{(r)}} \right), & \text{se } h+1 \leq v \leq s-1 \\ & \text{e } k_j^* < k_j^{(r)}. \end{cases} \tag{3.6}$$

Na Figura 3.6, para m grande, tanto o segmento limitado pelos pontos k_{h-1}^*+1 e $k_h^{(r)}$ quanto o bloco localizado entre os pontos k_{h-1}^*+1 à k_h^* , satisfazem igualdade (3.6), pois não existe nenhum ponto de independência dentro desses segmentos e ambos pertencem ao bloco limitado pelos pontos k_{h-1}^*+1 e k_h^* . Já o segmento que vai do ponto k_h^*+1 até $k_{h+1}^{(r+1)}$ também satisfaz (3.6), mas na iteração r não sabemos se existe outros pontos de independência antes de $k_{h+1}^{(r)}$, por isso deixamos a notação k_v^* com $h+1 \leq v \leq s-1$. Por fim usamos (3.7) no segmento entre os pontos $k_h^{(r)}+1$ e $k_{h+1}^{(r)}$, pois existe o ponto de independência k_h^* dentro do segmento. Assim com probabilidade 1

temos que

$$\begin{aligned}
\lim_{m \rightarrow \infty} l_m(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) &= \sum_{a_{k_{h-1}^*+1}^{k_h^{(r)}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \\
&\quad + \overbrace{\sum_{a_{k_h^{(r)}+1}^{k_h^{(r)}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \mathbb{P}_{\{k_h^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r)}} \right) \log \left[\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \mathbb{P}_{\{k_h^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r)}} \right) \right]}^{(1)} \\
&\quad - \sum_{a_{k_{h-1}^*+1}^{k_h^*}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \\
&\quad - \sum_{a_{k_h^*+1}^{k_{h+1}^{(r+1)}}} \mathbb{P}_{\{k_h^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r+1)}} \right) \log \mathbb{P}_{\{k_h^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r+1)}} \right). \tag{3.8}
\end{aligned}$$

Temos ainda $\lim_{m \rightarrow \infty} \frac{\log m}{m} = 0$, logo a penalidade tende a zero para m grande. Separando os somatórios podemos reescrever o termo (1) como

$$\begin{aligned}
&\left[\sum_{a_{k_h^*+1}^{k_{h+1}^{(r)}}} \mathbb{P}_{\{k_h^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r)}} \right) \right] \sum_{a_{k_h^{(r)}+1}^{k_h^*} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \\
&+ \left[\sum_{a_{k_h^{(r)}+1}^{k_h^*} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \right] \sum_{a_{k_h^*+1}^{k_{h+1}^{(r)}}} \mathbb{P}_{\{k_h^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r)}} \right) \log \mathbb{P}_{\{k_h^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r)}} \right) \tag{3.9}
\end{aligned}$$

por fim

$$\begin{aligned}
&\sum_{a_{k_h^{(r)}+1}^{k_h^*} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \\
&+ \sum_{a_{k_h^*+1}^{k_{h+1}^{(r)}}} \mathbb{P}_{\{k_h^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r)}} \right) \log \mathbb{P}_{\{k_h^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r)}} \right)
\end{aligned}$$

pois $\sum_{a_{k_h^*+1}^{k_{h+1}^{(r)}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) = 1$ e $\sum_{a_{k_h^*+1}^{k_{h+1}^{(r)}}} \mathbb{P}_{\{k_h^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r)}} \right) = 1$. Com isso a equação (3.8)

pode ser escrita como

$$\begin{aligned}
\lim_{m \rightarrow \infty} l_m(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) &= \sum_{\substack{k_h^{(r)} \\ a_{k_{h-1}^*+1}^{k_h^{(r)}}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \\
&+ \sum_{\substack{k_h^* \\ a_{k_h^{(r)}+1}^{k_h^*}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \\
&- \sum_{\substack{k_h^* \\ a_{k_{h-1}^*+1}^{k_h^*}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \quad (3.10)
\end{aligned}$$

Analisando o último termo de (3.10), podemos fazer

$$\sum_{\substack{k_h^* \\ a_{k_{h-1}^*+1}^{k_h^*}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \log \left[\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \frac{\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right)}{\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right)} \right] \quad (3.11)$$

separando os logaritmos temos

$$\begin{aligned}
&\overbrace{\sum_{\substack{k_h^* \\ a_{k_{h-1}^*+1}^{k_h^*}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right)}^{(2)} \\
&+ \overbrace{\sum_{\substack{k_h^* \\ a_{k_{h-1}^*+1}^{k_h^*}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \log \left(\frac{\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right)}{\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right)} \right)}^{(3)}.
\end{aligned}$$

Abrindo os somatórios do termo denotado por (2) temos

$$\sum_{\substack{k_h^{(r)} \\ a_{k_{h-1}^*+1}^{k_h^{(r)}}}} \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \sum_{\substack{k_h^* \\ a_{k_h^{(r)}+1}^{k_h^*}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \quad (3.12)$$

se olharmos o bloco entre os pontos $k_{h-1}^* + 1$ e k_h^* como uma distribuição conjunta dos segmentos referente aos pontos $k_{h-1}^* + 1$ e $k_h^{(r)}$ e $k_h^{(r)} + 1$ até k_h^* , temos que a soma em (3.12) é a distribuição

marginal em relação ao bloco k_{h-1}^* à $k_h^{(r)}$, portanto o termo (2) se torna

$$\sum_{a_{k_{h-1}^*+1}^{k_h^{(r)}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right). \quad (3.13)$$

Decompondo o somatório do termo (3) da mesma forma que na equação (3.12) temos

$$\sum_{a_{k_h^{(r)}+1}^{k_h^*} \left[\sum_{a_{k_{h-1}^*+1}^{k_h^{(r)}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \log \left(\frac{\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right)}{\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right)} \right) \right]$$

usando a Proposição A.2, ver apêndice, temos que

$$\begin{aligned} & \sum_{a_{k_{h-1}^*+1}^{k_h^{(r)}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \log \left(\frac{\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right)}{\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right)} \right) \\ & \geq \left(\sum_{a_{k_{h-1}^*+1}^{k_h^{(r)}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \right) \log \left(\frac{\sum_{a_{k_{h-1}^*+1}^{k_h^{(r)}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right)}{\sum_{a_{k_{h-1}^*+1}^{k_h^{(r)}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right)} \right) \\ & = \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \end{aligned}$$

já que $\sum_{a_{k_{h-1}^*+1}^{k_h^{(r)}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) = 1$. Usando a suposição que $k_h^{(r)} \neq k_h^*$ as probabilidades

$\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right)$ e $\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right)$ são diferentes para todos os valores de $a_{k_{h-1}^*+1}^{k_h^{(r)}} \in \mathcal{A}_h^{(r)}$. Com isso o termo (3) resulta em

$$\begin{aligned} & \sum_{a_{k_{h-1}^*+1}^{k_h^*} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \log \left(\frac{\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right)}{\mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right)} \right) \\ & > \sum_{a_{k_h^{(r)}+1}^{k_h^*} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^{(r)}+1}^{k_h^*} \right) \end{aligned} \quad (3.14)$$

Substituindo (3.13) e (3.14) na equação (3.10) obtemos

$$\begin{aligned}
\lim_{m \rightarrow \infty} l_m(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) &< \sum_{\substack{k_h^{(r)} \\ a_{k_{h-1}^*+1}^{k_h^{(r)}}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \\
&+ \sum_{\substack{k_h^* \\ a_{k_h^*+1}^{k_h^*}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^*+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^*+1}^{k_h^*} \right) \\
&- \sum_{\substack{k_h^{(r)} \\ a_{k_{h-1}^*+1}^{k_h^{(r)}}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \\
&- \sum_{\substack{k_h^* \\ a_{k_h^*+1}^{k_h^*}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^*+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^*+1}^{k_h^*} \right) \\
&= 0
\end{aligned}$$

ou seja,

$$LVP(\mathbf{k}^{(r)}|\mathbf{Y}) < LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}), \text{ quase certamente, quando } m \rightarrow \infty.$$

3.1.2 Caso em que $k_{h+w}^{(r)} < k_h^* < k_{h+w+1}^{(r)}$

Querendo atualizar o ponto $k_h^{(r)}$ e considerando ainda o vetor dado em (3.2) temos agora a situação quando o ponto de corte verdadeiro k_h^* está entre $k_{h+w}^{(r)}$ e $k_{h+w+1}^{(r)}$, ver Figura 3.7. Para esse caso temos que $k_{h+1}^{(r)} \neq k_{h+1}^*, \dots, k_{h+w}^{(r)} \neq k_{h+w}^*$ e esses pontos de corte serão ignorados quando atualizarmos o ponto k_h^* , fazendo com que o vetor $\mathbf{k}^{(r+1)}$ tenha w pontos de corte a menos do que $\mathbf{k}^{(r)}$.

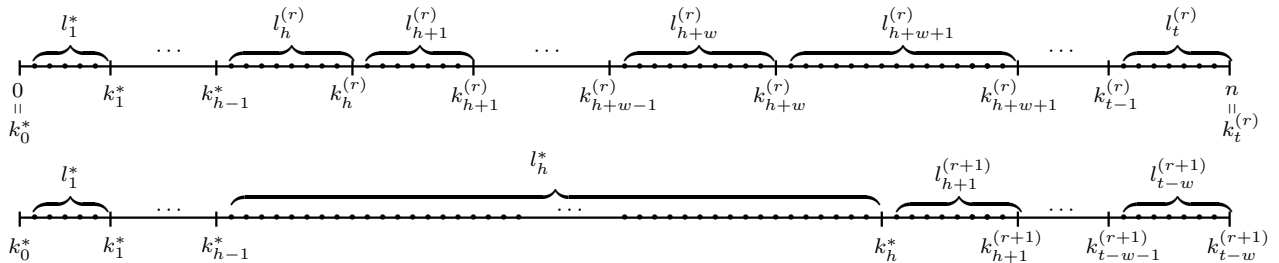


Figura 3.7: Ilustração dos vetores $\mathbf{k}^{(r)}$ e $\mathbf{k}^{(r+1)}$ em (3.2) quando $k_{h+w}^{(r)} < k_h^* < k_{h+w+1}^{(r)}$.

Assim definimos as componentes do vetor $\mathbf{k}^{(r+1)}$ da seguinte forma

$$\mathbf{k}_j^{(r+1)} = \begin{cases} k_j^*, & 1 \leq j \leq h \\ k_{j+w}^{(r)}, & h+1 \leq j \leq t-1-w \end{cases}$$

Neste caso o logaritmo da verossimilhança penalizada de $\mathbf{k}^{(r)}$ é da forma que em (3.4) e o logaritmo da verossimilhança penalizada do vetor atualizado $\mathbf{k}^{(r+1)}$ é dado por

$$\begin{aligned} LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}) &= \sum_{j=1}^h \sum_{\substack{k_j^* \\ a_{k_{j-1}^*+1}^*}} N_m^{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right) \log \left(\frac{N_m^{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right)}{m} \right) \\ &+ \sum_{\substack{k_{h+1}^{(r+1)} \\ a_{k_h^*+1}^*}} N_m^{\{k_h^*:k_{h+1}^{(r+1)}\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r+1)}} \right) \log \left(\frac{N_m^{\{k_h^*:k_{h+1}^{(r+1)}\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r+1)}} \right)}{m} \right) \\ &+ \sum_{j=h+2}^{t-w} \sum_{\substack{k_j^{(r+1)} \\ a_{k_{j-1}^{(r+1)}+1}}} N_m^{\{k_{j-1}^{(r+1)}:k_j^{(r+1)}\}} \left(a_{k_{j-1}^{(r+1)}+1}^{k_j^{(r+1)}} \right) \log \left(\frac{N_m^{\{k_{j-1}^{(r+1)}:k_j^{(r+1)}\}} \left(a_{k_{j-1}^{(r+1)}+1}^{k_j^{(r+1)}} \right)}{m} \right) \\ &- c \left(\sum_{j=1}^h |\mathcal{A}|^{l_j^*} + \sum_{j=h+1}^{t-w} |\mathcal{A}|^{l_j^{(r+1)}} - (t-w) \right) \log m. \end{aligned}$$

sendo $k_{h+1}^{(r+1)} = k_{h+w+1}^{(r)}, \dots, k_{t-w-1}^{(r+1)} = k_{t-1}^{(r)}$ e $k_{t-w}^{(r+1)} = k_t^{(r)} = n$, como mostrado na Figura 3.7.

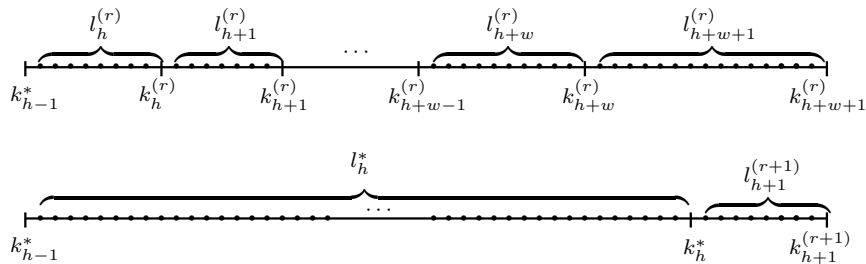


Figura 3.8: Atualização da iteração r para $r+1$, quando $k_{h+w}^{(r)} < k_h^* < k_{h+w+1}^{(r)}$.

Portanto a expressão (3.3) fica da seguinte forma

$$\begin{aligned}
l_m(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) &= \sum_{a_{k_{h-1}^*+1}^{k_h^{(r)}}} \hat{\mathbb{P}}_{\{k_{h-1}^*:k_h^{(r)}\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \log \hat{\mathbb{P}}_{\{k_{h-1}^*:k_h^{(r)}\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \\
&+ \sum_{j=h+1}^{h+w+1} \sum_{a_{k_{j-1}^*+1}^{k_j^{(r)}}} \hat{\mathbb{P}}_{\{k_{j-1}^*:k_j^{(r)}\}} \left(a_{k_{j-1}^*+1}^{k_j^{(r)}} \right) \log \hat{\mathbb{P}}_{\{k_{j-1}^*:k_j^{(r)}\}} \left(a_{k_{j-1}^*+1}^{k_j^{(r)}} \right) \\
&- \sum_{a_{k_{h-1}^*+1}^{k_h^*}} \hat{\mathbb{P}}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \log \hat{\mathbb{P}}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \\
&- \sum_{a_{k_h^*+1}^{k_{h+1}^{(r+1)}}} \hat{\mathbb{P}}_{\{k_h^*:k_{h+1}^{(r+1)}\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r+1)}} \right) \log \hat{\mathbb{P}}_{\{k_h^*:k_{h+1}^{(r+1)}\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r+1)}} \right) \\
&- c \left(\sum_{j=h}^{h+w+1} |\mathcal{A}|^{l_j^{(r)}} - |\mathcal{A}|^{l_h^*} - |\mathcal{A}|^{l_{h+1}^{(r+1)}} - w \right) \frac{\log m}{m}.
\end{aligned}$$

Para m suficientemente grande vemos que o segmento limitado pelos pontos $k_{h+w}^{(r)} + 1$ e $k_{h+w+1}^{(r)}$ satisfaz a igualdade (3.7), pois k_h^* é um ponto de independência e depois vamos separar o somatório usando a mesma idéia feita em (3.9). Vemos ainda que todos os outros segmentos da Figura 3.8 satisfazem a igualdade (3.6). Portanto

$$\begin{aligned}
\lim_{m \rightarrow \infty} l_m(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) &= \sum_{a_{k_{h-1}^*+1}^{k_h^{(r)}}} \mathbb{P}_{\{k_{h-1}^*:k_h^{(r)}\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^{(r)}\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \\
&+ \sum_{j=h+1}^{h+w} \sum_{a_{k_{j-1}^*+1}^{k_j^{(r)}}} \mathbb{P}_{\{k_{h-1}^*:k_h^{(r)}\}} \left(a_{k_{j-1}^*+1}^{k_j^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^{(r)}\}} \left(a_{k_{j-1}^*+1}^{k_j^{(r)}} \right) \\
&+ \sum_{a_{k_{h+w}^{(r)}+1}^{k_h^*}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h+w}^{(r)}+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h+w}^{(r)}+1}^{k_h^*} \right) \\
&+ \sum_{a_{k_h^*+1}^{k_{h+w+1}^{(r)}}} \mathbb{P}_{\{k_{h-1}^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+w+1}^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+w+1}^{(r)}} \right) \\
&- \sum_{a_{k_{h-1}^*+1}^{k_h^*}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \\
&- \sum_{a_{k_h^*+1}^{k_{h+1}^{(r+1)}}} \mathbb{P}_{\{k_h^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r+1)}} \right) \log \mathbb{P}_{\{k_h^*:k_v^*\}} \left(a_{k_h^*+1}^{k_{h+1}^{(r+1)}} \right)
\end{aligned}$$

com probabilidade 1.

$$\begin{aligned}
\lim_{m \rightarrow \infty} l_m(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) &= \sum_{\substack{k_h^{(r)} \\ a_{k_{h-1}^*+1}^{k_h^*}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^{(r)}} \right) \\
&+ \sum_{j=h+1}^{h+w} \sum_{\substack{k_j^{(r)} \\ a_{k_{j-1}^*+1}^{k_j^*}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{j-1}^*+1}^{k_j^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{j-1}^*+1}^{k_j^{(r)}} \right) \\
&+ \sum_{\substack{k_h^* \\ a_{k_{h+w}^*+1}^{k_h^{(r)}}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h+w}^*+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h+w}^*+1}^{k_h^*} \right) \\
&- \sum_{\substack{k_h^* \\ a_{k_{h-1}^*+1}^{k_h^*}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h-1}^*+1}^{k_h^*} \right) \quad (3.15)
\end{aligned}$$

aplicando a mesma idéia em (3.11) no último termo de (3.15) obtemos as expressões (3.13) e (3.14), assim

$$\begin{aligned}
\lim_{m \rightarrow \infty} l_m(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) &< \sum_{j=h+1}^{h+w} \sum_{\substack{k_j^{(r)} \\ a_{k_{j-1}^*+1}^{k_j^*}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{j-1}^*+1}^{k_j^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{j-1}^*+1}^{k_j^{(r)}} \right) \\
&+ \sum_{\substack{k_h^* \\ a_{k_{h+w}^*+1}^{k_h^{(r)}}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h+w}^*+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h+w}^*+1}^{k_h^*} \right) \\
&- \sum_{\substack{k_h^* \\ a_{k_h^*+1}^{k_h^{(r)}}}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^*+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_h^*+1}^{k_h^*} \right). \quad (3.16)
\end{aligned}$$

Vamos repetir o mesmo procedimento outras w vezes, obtemos

$$\begin{aligned}
\lim_{m \rightarrow \infty} l_m(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) &< \sum_{j=h+1}^{h+w} \sum_{\substack{k_j^{(r)} \\ a_{k_{j-1}^{(r)}+1}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \\
&+ \sum_{\substack{k_h^* \\ a_{k_{h+w}^{(r)}+1}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h+w}^{(r)}+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h+w}^{(r)}+1}^{k_h^*} \right) \\
&- \sum_{j=h+1}^{h+w} \sum_{\substack{k_j^{(r)} \\ a_{k_{j-1}^{(r)}+1}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \\
&- \sum_{\substack{k_h^* \\ a_{k_{h+w}^{(r)}+1}} \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h+w}^{(r)}+1}^{k_h^*} \right) \log \mathbb{P}_{\{k_{h-1}^*:k_h^*\}} \left(a_{k_{h+w}^{(r)}+1}^{k_h^*} \right) \\
&= 0
\end{aligned}$$

com isso

$$LVP(\mathbf{k}^{(r)}|\mathbf{Y}) < LVP(\mathbf{k}^{(r+1)}|\mathbf{Y})$$

quase certamente, quando $m \rightarrow \infty$.

3.1.3 Caso em que $\mathbf{k}^{(r)} = (k_1^*, \dots, k_{s-1}^*, k_s^{(r)}, \dots, k_{t-1}^{(r)})$

Agora vamos demonstrar a situação em que estimamos $t > s$ pontos de corte, sendo que os primeiros $s - 1$ pontos de corte do modelo estão nas suas posições corretas. Iremos eliminar os demais pontos de corte a cada iteração. O vetor $\mathbf{k}^{(r+1)}$ pode ser escrito da seguinte forma

$$\mathbf{k}_j^{(r+1)} = \begin{cases} k_j^*, & 1 \leq j \leq s-1 \\ k_{j+1}^{(r)}, & s \leq j \leq t-2 \end{cases} \quad (3.17)$$

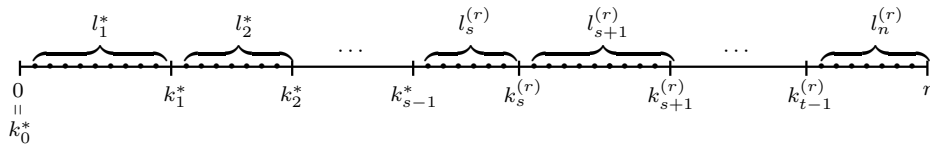


Figura 3.9: Esboço de um possível vetor $\mathbf{k}^{(r)} = (k_1^*, \dots, k_{s-1}^*, k_s^{(r)}, \dots, k_{t-1}^{(r)})$.

Nessa situação o logaritmo da verossimilhança penalizada de $\mathbf{k}^{(r)}$ dado \mathbf{Y} , ver Figura 3.9, é dado por

$$\begin{aligned}
LVP(\mathbf{k}^{(r)}|\mathbf{Y}) &= \sum_{j=1}^{s-1} \sum_{\substack{k_j^* \\ a_{k_{j-1}^*+1}}} N_m^{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right) \log \left(\frac{N_m^{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right)}{m} \right) \\
&+ \sum_{\substack{k_s^{(r)} \\ a_{k_{s-1}^*+1}}} N_m^{\{k_{s-1}^*:k_s^{(r)}\}} \left(a_{k_{s-1}^*+1}^{k_s^{(r)}} \right) \log \left(\frac{N_m^{\{k_{s-1}^*:k_s^{(r)}\}} \left(a_{k_{s-1}^*+1}^{k_s^{(r)}} \right)}{m} \right) \\
&+ \sum_{j=s+1}^t \sum_{\substack{k_j^{(r)} \\ a_{k_{j-1}^{(r)}+1}}} N_m^{\{k_{j-1}^{(r)}:k_j^{(r)}\}} \left(a_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \log \left(\frac{N_m^{\{k_{j-1}^{(r)}:k_j^{(r)}\}} \left(a_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right)}{m} \right) \\
&+ \sum_{\substack{k_{t-1}^{(r)} \\ a_{k_{t-1}^{(r)}+1}}} N_m^{\{k_{t-1}^{(r)}:n\}} \left(a_{k_{t-1}^{(r)}+1}^n \right) \log \left(\frac{N_m^{\{k_{t-1}^{(r)}:n\}} \left(a_{k_{t-1}^{(r)}+1}^n \right)}{m} \right) \\
&- c \left(\sum_{j=1}^{s-1} |\mathcal{A}|^{l_j^*} + \sum_{j=s}^{t-1} |\mathcal{A}|^{l_j^{(r)}} + |\mathcal{A}|^{l_n^{(r)}} - t \right) \log m. \tag{3.18}
\end{aligned}$$

Já o logaritmo da função de verossimilhança do vetor atualizado $\mathbf{k}^{(r+1)}$, como mostrado na Figura 3.10, é

$$\begin{aligned}
LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}) &= \sum_{j=1}^{s-1} \sum_{\substack{k_j^* \\ a_{k_{j-1}^*+1}}} N_m^{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right) \log \left(\frac{N_m^{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right)}{m} \right) \\
&+ \sum_{\substack{k_s^{(r+1)} \\ a_{k_{s-1}^*+1}}} N_m^{\{k_{s-1}^*:k_s^{(r+1)}\}} \left(a_{k_{s-1}^*+1}^{k_s^{(r+1)}} \right) \log \left(\frac{N_m^{\{k_{s-1}^*:k_s^{(r+1)}\}} \left(a_{k_{s-1}^*+1}^{k_s^{(r+1)}} \right)}{m} \right) \\
&+ \sum_{j=s+1}^{t-2} \sum_{\substack{k_j^{(r+1)} \\ a_{k_{j-1}^{(r+1)}+1}}} N_m^{\{k_{j-1}^{(r+1)}:k_j^{(r+1)}\}} \left(a_{k_{j-1}^{(r+1)}+1}^{k_j^{(r+1)}} \right) \log \left(\frac{N_m^{\{k_{j-1}^{(r+1)}:k_j^{(r+1)}\}} \left(a_{k_{j-1}^{(r+1)}+1}^{k_j^{(r+1)}} \right)}{m} \right) \\
&+ \sum_{\substack{k_{t-2}^{(r+1)} \\ a_{k_{t-2}^{(r+1)}+1}}} N_m^{\{k_{t-2}^{(r+1)}:n\}} \left(a_{k_{t-2}^{(r+1)}+1}^n \right) \log \left(\frac{N_m^{\{k_{t-2}^{(r+1)}:n\}} \left(a_{k_{t-2}^{(r+1)}+1}^n \right)}{m} \right) \\
&- c \left(\sum_{j=1}^{s-1} |\mathcal{A}|^{l_j^*} + \sum_{j=s}^{t-2} |\mathcal{A}|^{l_j^{(r+1)}} + |\mathcal{A}|^{l_n^{(r+1)}} - (t-1) \right) \log m. \tag{3.19}
\end{aligned}$$

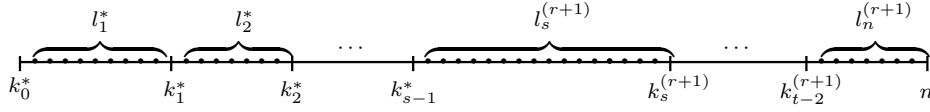


Figura 3.10: Ilustração do vetor $\mathbf{k}^{(r+1)}$ em (3.17).

Fazendo a diferença dos termos (3.18) e (3.19) obtemos

$$\begin{aligned}
\ell_m(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) &= \sum_{a_{k_{s-1}^*+1}^{k_s^{(r)}}} \hat{\mathbb{P}}_{\{k_{s-1}^*:k_s^{(r)}\}} \left(a_{k_{s-1}^*+1}^{k_s^{(r)}} \right) \log \hat{\mathbb{P}}_{\{k_{s-1}^*:k_s^{(r)}\}} \left(a_{k_{s-1}^*+1}^{k_s^{(r)}} \right) \\
&+ \sum_{a_{k_s^{(r)}+1}^{k_{s+1}^{(r)}}} \hat{\mathbb{P}}_{\{k_s^{(r)}:k_{s+1}^{(r)}\}} \left(a_{k_s^{(r)}+1}^{k_{s+1}^{(r)}} \right) \log \hat{\mathbb{P}}_{\{k_s^{(r)}:k_{s+1}^{(r)}\}} \left(a_{k_s^{(r)}+1}^{k_{s+1}^{(r)}} \right) \\
&- \sum_{a_{k_{s-1}^*+1}^{k_s^{(r+1)}}} \hat{\mathbb{P}}_{\{k_{s-1}^*:k_s^{(r+1)}\}} \left(a_{k_{s-1}^*+1}^{k_s^{(r+1)}} \right) \log \hat{\mathbb{P}}_{\{k_{s-1}^*:k_s^{(r+1)}\}} \left(a_{k_{s-1}^*+1}^{k_s^{(r+1)}} \right) \\
&- c \left(|\mathcal{A}|^{l_s^{(r)}} + |\mathcal{A}|^{l_{s+1}^{(r)}} - |\mathcal{A}|^{l_s^{(r+1)}} - 1 \right) \frac{\log m}{m}
\end{aligned}$$

ver ilustração na Figura 3.11.

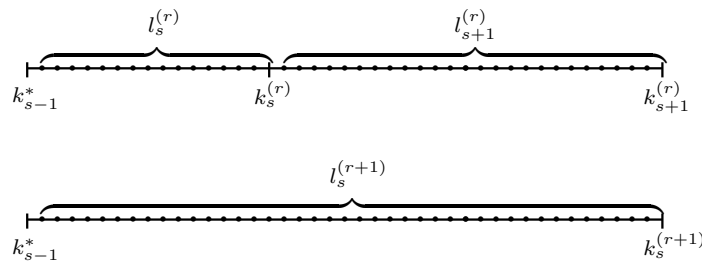


Figura 3.11: Remoção do ponto $k_s^{(r)}$ na iteração r para $r+1$.

Para m grande e aplicando (3.6) temos que

$$\begin{aligned} \lim_{m \rightarrow \infty} \ell_m(\mathbf{k}^{(r)}, \mathbf{k}^{(r+1)}) &= \sum_{a_{k_{s-1}^*+1}^{k_s^{(r)}}} \mathbb{P}_{\{k_{s-1}^*:n\}} \left(a_{k_{s-1}^*+1}^{k_s^{(r)}} \right) \log \mathbb{P}_{\{k_{s-1}^*:n\}} \left(a_{k_{s-1}^*+1}^{k_s^{(r)}} \right) \\ &+ \sum_{a_{k_s^{(r)}+1}^{k_{s+1}^{(r)}}} \mathbb{P}_{\{k_{s-1}^*:n\}} \left(a_{k_s^{(r)}+1}^{k_{s+1}^{(r)}} \right) \log \mathbb{P}_{\{k_{s-1}^*:n\}} \left(a_{k_s^{(r)}+1}^{k_{s+1}^{(r)}} \right) \\ &- \sum_{a_{k_{s-1}^*+1}^{k_s^{(r+1)}}} \mathbb{P}_{\{k_{s-1}^*:n\}} \left(a_{k_{s-1}^*+1}^{k_s^{(r+1)}} \right) \log \mathbb{P}_{\{k_{s-1}^*:n\}} \left(a_{k_{s-1}^*+1}^{k_s^{(r+1)}} \right) \end{aligned}$$

quase certamente. Note que esse limite tem a mesma estrutura que em (3.10) e a demonstração segue com os mesmos argumentos que na subseção 3.1.1.

3.1.4 Caso em que $\mathbf{k}^{(r)} = (k_1^*, \dots, k_{t-1}^*)$.

Nesta subseção temos que analisar o caso em que estimamos um número $t < s$ de pontos de corte, sendo necessário adicionar outros pontos de corte e verificar em qual posição a verossimilhança penalizada é máxima, até obter o vetor \mathbf{k}^* . Este será o caso em que a penalidade terá um papel importante. Primeiro vamos definir as componentes do vetor $\mathbf{k}^{(r+1)}$ como

$$k_j^{(r+1)} = k_j^*, \quad 1 \leq j \leq t < s. \quad (3.20)$$

Podemos escrever o logaritmo da verossimilhança penalizada de $\mathbf{k}^{(r)}$ dado \mathbf{Y} , ver Figura 3.12, da seguinte forma

$$\begin{aligned} LVP(\mathbf{k}^{(r)}|\mathbf{Y}) &= \sum_{j=1}^{t-1} \sum_{a_{k_{j-1}^*+1}^{k_j^*}} N_m^{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right) \log \left(\frac{N_m^{\{k_{j-1}^*:k_j^*\}} \left(a_{k_{j-1}^*+1}^{k_j^*} \right)}{m} \right) \\ &+ \sum_{a_{k_{t-1}^*+1}^n} N_m^{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right) \log \left(\frac{N_m^{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right)}{m} \right) \\ &- c \left(\sum_{j=1}^{t-1} |\mathcal{A}|^{l_j^*} + |\mathcal{A}|^{l_n^{(r)}} - t \right) \log m \end{aligned}$$

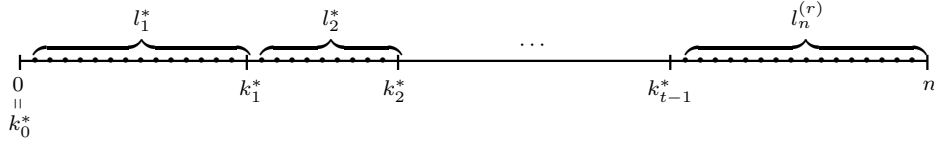


Figura 3.12: Ilustração do vetor $\mathbf{k}^{(r)} = (k_1^*, \dots, k_{t-1}^*)$.

e o logaritmo da verossimilhança penalizada de $\mathbf{k}^{(r+1)}$ dado \mathbf{Y} é da forma

$$\begin{aligned} LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}) &= \sum_{j=1}^t \sum_{\substack{k_j^* \\ a_{k_{j-1}^*+1}^{k_j^*}}} N_m^{\{k_{j-1}^*:k_j^*\}}(a_{k_{j-1}^*+1}^{k_j^*}) \log \left(\frac{N_m^{\{k_{j-1}^*:k_j^*\}}(a_{k_{j-1}^*+1}^{k_j^*})}{m} \right) \\ &+ \sum_{a_{k_t^*+1}^n} N_m^{\{k_t^*:n\}}(a_{k_t^*+1}^n) \log \left(\frac{N_m^{\{k_t^*:n\}}(a_{k_t^*+1}^n)}{m} \right) \\ &- c \left(\sum_{j=1}^t |\mathcal{A}|^{l_j^*} + |\mathcal{A}|^{l_n^{(r+1)}} - (t+1) \right) \log m. \end{aligned}$$

ver esboço na Figura 3.13.

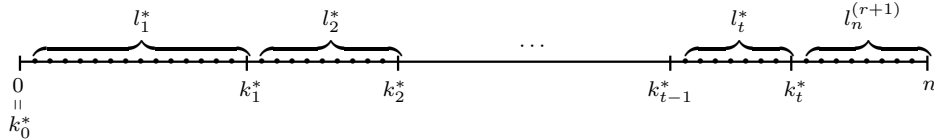


Figura 3.13: Ilustração do vetor $\mathbf{k}^{(r+1)}$ em (3.20).

Podemos escrever a diferença entre $LVP(\mathbf{k}^{(r)})$ e $LVP(\mathbf{k}^{(r+1)})$ da seguinte forma

$$\begin{aligned} LVP(\mathbf{k}^{(r)}|\mathbf{Y}) - LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}) &= \sum_{a_{k_{t-1}^*+1}^n} N_m^{\{k_{t-1}^*:n\}}(a_{k_{t-1}^*+1}^n) \log \hat{\mathbb{P}}_{\{k_{t-1}^*:n\}}(a_{k_{t-1}^*+1}^n) \\ &- \sum_{a_{k_{t-1}^*+1}^{k_t^*}} N_m^{\{k_{t-1}^*:k_t^*\}}(a_{k_{t-1}^*+1}^{k_t^*}) \log \hat{\mathbb{P}}_{\{k_{t-1}^*:k_t^*\}}(a_{k_{t-1}^*+1}^{k_t^*}) \\ &- \sum_{a_{k_t^*+1}^n} N_m^{\{k_t^*:n\}}(a_{k_t^*+1}^n) \log \hat{\mathbb{P}}_{\{k_t^*:n\}}(a_{k_t^*+1}^n) \\ &- c \left(|\mathcal{A}|^{l_n^{(r)}} - |\mathcal{A}|^{l_t^*} - |\mathcal{A}|^{l_n^{(r+1)}} + 1 \right) \log m \end{aligned}$$

(ver Figura 3.14) somando e subtraindo o termo $\sum_{a_{k_{t-1}^*+1}^n} N_m^{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right) \log \mathbb{P}_{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right)$ e usando a independência do ponto k_t^* temos que

$$\begin{aligned}
LVP(\mathbf{k}^{(r)}|\mathbf{Y}) - LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}) &= \sum_{a_{k_{t-1}^*+1}^n} N_m^{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right) \log \frac{\hat{\mathbb{P}}_{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right)}{\mathbb{P}_{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right)} \\
&\quad \underbrace{\hspace{10em}}_{(4)} \\
&+ \sum_{a_{k_{t-1}^*+1}^n} N_m^{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right) \log \mathbb{P}_{\{k_{t-1}^*:k_t^*\}} \left(a_{k_{t-1}^*+1}^{k_t^*} \right) \mathbb{P}_{\{k_{t-1}^*:k_t^*\}} \left(a_{k_t^*+1}^n \right) \\
&- \sum_{a_{k_{t-1}^*+1}^{k_t^*}} N_m^{\{k_{t-1}^*:k_t^*\}} \left(a_{k_{t-1}^*+1}^{k_t^*} \right) \log \hat{\mathbb{P}}_{\{k_{t-1}^*:k_t^*\}} \left(a_{k_{t-1}^*+1}^{k_t^*} \right) \\
&- \sum_{a_{k_t^*+1}^n} N_m^{\{k_t^*:n\}} \left(a_{k_t^*+1}^n \right) \log \hat{\mathbb{P}}_{\{k_t^*:n\}} \left(a_{k_t^*+1}^n \right) \\
&- c \left(|\mathcal{A}|^{l_n^{(r)}} - |\mathcal{A}|^{l_t^*} - |\mathcal{A}|^{l_n^{(r+1)}} + 1 \right) \log m
\end{aligned}$$

separando o logaritmo do termo (4) obtemos

$$\begin{aligned}
LVP(\mathbf{k}^{(r)}|\mathbf{Y}) - LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}) &= \sum_{a_{k_{t-1}^*+1}^n} N_m^{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right) \log \frac{\hat{\mathbb{P}}_{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right)}{\mathbb{P}_{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right)} \\
&+ \sum_{a_{k_{t-1}^*+1}^{k_t^*}} \log \mathbb{P}_{\{k_{t-1}^*:k_t^*\}} \left(a_{k_{t-1}^*+1}^{k_t^*} \right) \sum_{a_{k_t^*+1}^n} N_m^{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right) \\
&+ \sum_{a_{k_t^*+1}^n} \log \mathbb{P}_{\{k_{t-1}^*:k_t^*\}} \left(a_{k_t^*+1}^n \right) \sum_{a_{k_{t-1}^*+1}^{k_t^*}} N_m^{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right) \\
&- \sum_{a_{k_{t-1}^*+1}^{k_t^*}} N_m^{\{k_{t-1}^*:k_t^*\}} \left(a_{k_{t-1}^*+1}^{k_t^*} \right) \log \hat{\mathbb{P}}_{\{k_{t-1}^*:k_t^*\}} \left(a_{k_{t-1}^*+1}^{k_t^*} \right) \\
&- \sum_{a_{k_t^*+1}^n} N_m^{\{k_t^*:n\}} \left(a_{k_t^*+1}^n \right) \log \hat{\mathbb{P}}_{\{k_t^*:n\}} \left(a_{k_t^*+1}^n \right) \\
&- c \left(|\mathcal{A}|^{l_n^{(r)}} - |\mathcal{A}|^{l_t^*} - |\mathcal{A}|^{l_n^{(r+1)}} + 1 \right) \log m \tag{3.21}
\end{aligned}$$

sabemos que $\sum_{a_{k_{t-1}^*+1}^n} N_m^{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right) = N_m^{\{k_{t-1}^*:k_t^*\}} \left(a_{k_{t-1}^*+1}^{k_t^*} \right) e \sum_{a_{k_{t-1}^*+1}^{k_t^*}} N_m^{\{k_{t-1}^*:n\}} \left(a_{k_{t-1}^*+1}^n \right) =$

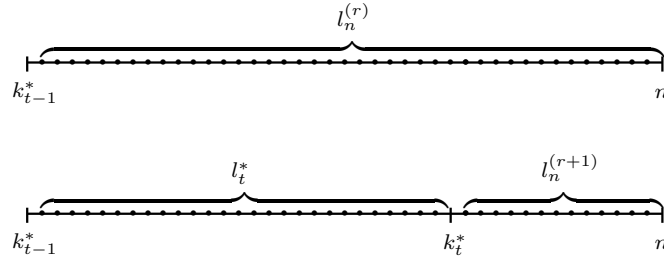


Figura 3.14: Adição do ponto k_t^* da iteração r para $r + 1$.

$N_m^{\{k_t^*:n\}}(a_{k_t^*+1}^n)$, assim podemos reescrever o termo (3.21) da seguinte maneira

$$\begin{aligned}
LVP(\mathbf{k}^{(r)}|\mathbf{Y}) - LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}) &= \sum_{a_{k_{t-1}^*+1}^n} N_m^{\{k_{t-1}^*:n\}}(a_{k_{t-1}^*+1}^n) \log \frac{\hat{\mathbb{P}}_{\{k_{t-1}^*:n\}}(a_{k_{t-1}^*+1}^n)}{\mathbb{P}_{\{k_{t-1}^*:n\}}(a_{k_{t-1}^*+1}^n)} \\
&- \sum_{a_{k_{t-1}^*+1}^{k_t^*}} N_m^{\{k_{t-1}^*:k_t^*\}}(a_{k_{t-1}^*+1}^{k_t^*}) \log \frac{\hat{\mathbb{P}}_{\{k_{t-1}^*:k_t^*\}}(a_{k_{t-1}^*+1}^{k_t^*})}{\mathbb{P}_{\{k_{t-1}^*:k_t^*\}}(a_{k_{t-1}^*+1}^{k_t^*})} \\
&- \sum_{a_{k_t^*+1}^n} N_m^{\{k_t^*:n\}}(a_{k_t^*+1}^n) \log \frac{\hat{\mathbb{P}}_{\{k_t^*:n\}}(a_{k_t^*+1}^n)}{\mathbb{P}_{\{k_t^*:n\}}(a_{k_t^*+1}^n)} \\
&- c(|\mathcal{A}|^{l_n^{(r)}} - |\mathcal{A}|^{l_t^*} - |\mathcal{A}|^{l_n^{(r+1)}} + 1) \log m
\end{aligned}$$

usando a Definição A.1 do Apêndice A, temos que

$$\begin{aligned}
LVP(\mathbf{k}^{(r)}|\mathbf{Y}) - LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}) &= mD \left(\hat{\mathbb{P}}_{\{k_{t-1}^*:n\}}(a_{k_{t-1}^*+1}^n) \middle\| \mathbb{P}_{\{k_{t-1}^*:n\}}(a_{k_{t-1}^*+1}^n) \right) \\
&\quad \quad \quad (5) \\
&- mD \left(\hat{\mathbb{P}}_{\{k_{t-1}^*:k_t^*\}}(a_{k_{t-1}^*+1}^{k_t^*}) \middle\| \mathbb{P}_{\{k_{t-1}^*:k_t^*\}}(a_{k_{t-1}^*+1}^{k_t^*}) \right) \\
&\quad \quad \quad (6) \\
&- mD \left(\hat{\mathbb{P}}_{\{k_t^*:n\}}(a_{k_t^*+1}^n) \middle\| \mathbb{P}_{\{k_t^*:n\}}(a_{k_t^*+1}^n) \right) \\
&- c(|\mathcal{A}|^{l_n^{(r)}} - |\mathcal{A}|^{l_t^*} - |\mathcal{A}|^{l_n^{(r+1)}} + 1) \log m
\end{aligned}$$

aplicando a Proposição 4.2 nos termos (5) e (6) obtemos que

$$mD\left(\hat{\mathbb{P}}_{\{k_{t-1}^*:k_t^*\}}\left(a_{k_{t-1}^*+1}^{k_t^*}\right)\left\|\mathbb{P}_{\{k_{t-1}^*:k_t^*\}}\left(a_{k_{t-1}^*+1}^{k_t^*}\right)\right.\right) \geq 0.$$

e

$$mD\left(\hat{\mathbb{P}}_{\{k_t^*:n\}}\left(a_{k_t^*+1}^n\right)\left\|\mathbb{P}_{\{k_t^*:n\}}\left(a_{k_t^*+1}^n\right)\right.\right) \geq 0.$$

Com isso temos que

$$\begin{aligned} LVP(\mathbf{k}^{(r)}|\mathbf{Y}) - LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}) &\leq mD\left(\hat{\mathbb{P}}_{\{k_{t-1}^*:n\}}\left(a_{k_{t-1}^*+1}^n\right)\left\|\mathbb{P}_{\{k_{t-1}^*:n\}}\left(a_{k_{t-1}^*+1}^n\right)\right.\right) \\ &\quad - c\left(|\mathcal{A}|^{l_n^{(r)}} - |\mathcal{A}|^{l_t^*} - |\mathcal{A}|^{l_n^{(r+1)}} + 1\right) \log m \end{aligned} \quad (3.22)$$

usando o lema 4.3 e em seguida a proposição 4.4, para algum $\delta > 0$, temos que

$$\begin{aligned} mD\left(\hat{\mathbb{P}}_{\{k_{t-1}^*:n\}}\left(a_{k_{t-1}^*+1}^n\right)\left\|\mathbb{P}_{\{k_{t-1}^*:n\}}\left(a_{k_{t-1}^*+1}^n\right)\right.\right) &\leq m \sum_{a_{k_{t-1}^*+1}^n} \frac{\left[\hat{\mathbb{P}}_{\{k_{t-1}^*:n\}}\left(a_{k_{t-1}^*+1}^n\right) - \mathbb{P}_{\{k_{t-1}^*:n\}}\left(a_{k_{t-1}^*+1}^n\right)\right]^2}{\mathbb{P}_{\{k_{t-1}^*:n\}}\left(a_{k_{t-1}^*+1}^n\right)} \\ &< \frac{\delta \log m}{\mathbb{P}_{\{k_{t-1}^*:n\}}\left(a_{k_{t-1}^*+1}^n\right)} \end{aligned}$$

quase certamente, quando $m \rightarrow \infty$, agora podemos escrever (3.22) da seguinte forma

$$LVP(\mathbf{k}^{(r)}|\mathbf{Y}) - LVP(\mathbf{k}^{(r+1)}|\mathbf{Y}) < \left[\frac{\delta}{\mathbb{P}_{min}} - c\left(|\mathcal{A}|^{l_n^{(r)}} - |\mathcal{A}|^{l_t^*} - |\mathcal{A}|^{l_n^{(r+1)}} + 1\right)\right] \log m$$

sendo

$$\mathbb{P}_{min} = \min\left\{\mathbb{P}_{\{k_{t-1}^*:n\}}\left(a_{k_{t-1}^*+1}^n\right) : \mathbb{P}_{\{k_{t-1}^*:n\}}\left(a_{k_{t-1}^*+1}^n\right) > 0, \text{ com } a_{k_{t-1}^*+1}^n \in \mathcal{A}^{l_n^{(r)}}\right\}$$

um minorante para $\mathbb{P}_{\{k_{t-1}^*:n\}}\left(a_{k_{t-1}^*+1}^n\right)$. Adotando $\delta < c\mathbb{P}_{min}\left(|\mathcal{A}|^{l_n^{(r)}} - |\mathcal{A}|^{l_t^*} - |\mathcal{A}|^{l_n^{(r+1)}} + 1\right)$ fica mostrado que

$$LVP(\mathbf{k}^{(r)}|\mathbf{Y}) < LVP(\mathbf{k}^{(r+1)}|\mathbf{Y})$$

quase certamente, quando $m \rightarrow \infty$. Por fim, com as demonstrações nas subseções 3.1.1, 3.1.2, 3.1.3 e 3.1.4 fica provado o Lema 2.3.1.

3.2 Demonstração do Teorema 2.3.2

Na demonstração do Lema 2.3.1 foi mostrado que, dado um vetor qualquer de pontos de corte, sempre podemos melhorar as suas estimativas de forma que o logaritmo da verossimilhança penalizada do vetor $\mathbf{k}^{(r+1)}$ é maior que o obtido no vetor $\mathbf{k}^{(r)}$. Dessa forma a demonstração do Teorema 2.3.2 segue por consequência do Lema 2.3.1, já que dado um vetor qualquer de pontos de corte estimados, para m grande e em um número finito de iterações obteremos o vetor \mathbf{k}^* , como ilustrado nas Figuras 3.1 e 3.2. Dessa forma fica mostrada a consistência forte do estimador $\hat{\mathbf{k}}_m$ na definição 2.2.1.

Capítulo 4

Simulações

Neste capítulo apresentamos o algoritmo para segmentar sequências baseado na técnica Dividir e Conquistar e estudamos através de simulações o comportamento do modelo em diferentes situações. Mostramos também o que acontece quando mudamos o tamanho da amostra m . Os códigos fonte das simulações, executáveis no software R, nos Exemplos [4.2.1](#) e [4.2.2](#) estão disponíveis no Apêndice B.

4.1 Algoritmo de Estimação

A fim de obter um número ótimo de segmentos propomos o seguinte algoritmo:

- i) Calcular a verossimilhança penalizada da sequência de tamanho n , sem nenhum ponto de corte;
- ii) Considerar o modelo com dois segmentos. Se a máxima verossimilhança penalizada para esse caso for menor que a obtida no passo **i**, o processo é interrompido (critério de parada) e teremos somente 1 segmento. Caso contrário, devemos identificar qual será o ponto de corte estimado retorna o valor de máxima verossimilhança penalizada e ver próximo passo;
- iii) Repetir o passo **ii** em cada segmento formado até acontecer o critério de parada em todos eles.

4.2 Exemplos

Exemplo 4.2.1. Suponha uma amostra dos mesmos segmentos apresentadas no Exemplo [2.2.2](#) com a diferença que o verdadeiro ponto de corte é $k_1^* = 4$, $m = 2000$ e $n = 10$, ver uma representação na Figura [1.1](#). Executamos o algoritmo e a partir da amostra simulada vimos que o ponto de corte estimado $k_1 = 4$ possui maior verossimilhança penalizada, ver Figura [4.1](#).

Na Tabela 4.1, analisamos a necessidade de inserir ou não um ponto de corte, em cada segmento, pelo logaritmo da máxima verossimilhança penalizada (LMVP). Assim, no segmento de 1 a n vemos que o LMVP é maior quando estimamos um ponto de corte. Agora analisando os segmentos limitado pelos pontos 1 a k_1 e $k_1 + 1$ a n vemos que não há necessidade de adicionar um ponto de corte.

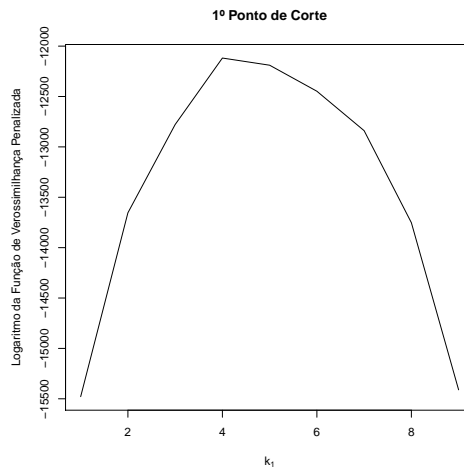


Figura 4.1: Gráfico do logaritmo da função de verossimilhança penalizada em relação ao ponto de corte k_1 quando existem 2 segmentos no Exemplo 4.2.1.

Segmento	Nº de pontos de cortes estimados	Logaritmo da máxima verossimilhança penalizada
1 a n	0	-18894,630
	1	-12118,310
1 a k_1	0	-4834,639
	1	-5029,113
k_1+1 a n	0	-7283,668
	1	-7332,187

Tabela 4.1: Logaritmo da função de verossimilhança penalizada para diferentes número de segmentos no Exemplo 4.2.1.

Exemplo 4.2.2. Suponha uma amostra de $m = 20000$ seqüências aleatórias $Y \in \mathcal{A}^{12}$, isto é $n = 12$, composta de 3 segmentos aleatórios independentes X_1 , X_2 e X_3 , sendo X_1 e X_2 iguais ao apresentado no Exemplo 2.2.2 e o segmento X_3 é gerado a partir de uma cadeia de Markov de alcance variável de ordem 2, ver Figura 4.2. Nesse exemplo denotamos os verdadeiros pontos de corte por $k_1^* = 4$ e $k_2^* = 9$. Na Tabela 4.2 vemos a necessidade de adicionar um ponto de corte nos segmentos limitado pelos pontos 1 a n e $k_1 + 1$ a n , já que ambos retornam maior LMVP quando o número de pontos de corte estimados é 1. É mostrado na Figura 4.3 (a) que o ponto de

corte estimado $k_1 = 4$ possui maior verossimilhança penalizada, em seguida analisando o segmento referente os pontos 1 a 3 e o segmento limitado entre os pontos 5 e 11 vemos que a verossimilhança penalizada é maior quando adotamos os pontos $k_1 = 4$ e $k_2 = 9$, ver Figura 4.3 (b).

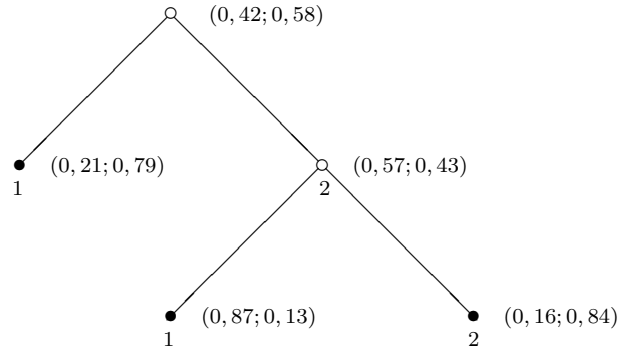


Figura 4.2: *Árvore de Contexto obtida a partir do segmento aleatório X_3 .*

Segmento	Nº de pontos de cortes estimados	Logaritmo da máxima verossimilhança penalizada
1 a n	0	-177465,90
	1	-141328,70
1 a k_1	0	-47189,70
	1	-49835,78
k_1+1 a n	0	-94138,95
	1	-92340,29
k_1+1 a k_2	0	-56985,36
	1	-60038,47
k_2+1 a n	0	-35110,59
	1	-38573,72

Tabela 4.2: *Logaritmo da função de verossimilhança penalizada para diferentes número de segmentos no Exemplo 4.2.2.*

Vamos estudar agora o que acontece quando a amostra não é muito grande. Perceba que quanto maior for a sequência maior deverá ser a amostra, pois quando realizamos uma segmentação devemos considerar qualquer ponto como um possível candidato a ponto de corte. Assim quanto maior for o tamanho do segmento menor será a chance de alguma palavra se repetir, nesse caso uma amostra muito grande resolveria esse problema. Dessa forma, para m pequeno, o algoritmo retorna como candidato o ponto de corte $\lfloor \frac{l}{2} \rfloor$. Temos que realizar esse processo até obter vários pequenos segmentos, assim podemos analisá-los e obter outros pontos de corte que não seja o valor central do segmento. Na Figura 4.4 foi fixado $n = 14$ temos simulações dos segmentos X_1 e X_2

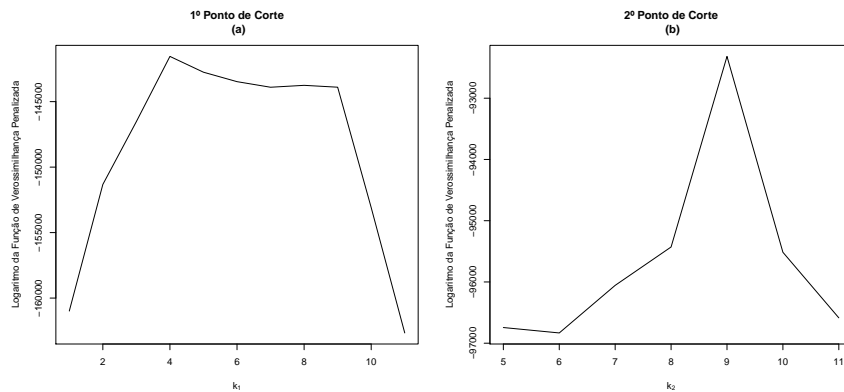


Figura 4.3: Gráfico do logaritmo da função de verossimilhança penalizada em relação ao ponto de corte nos casos com 2 e 3 segmentos, respectivamente.

definidos anteriormente e com ponto de corte verdadeiro $k_1^* = 10$ e vemos como se comporta a função de verossimilhança penalizada quando m cresce. Em (a) e (b) vemos que praticamente não houve alteração na forma da verossimilhança penalizada e o ponto de corte estimado é $k_1 = 7$, em (c) vemos um leve aumento no ponto 10 e em (d) temos que o ponto de corte se torna $k_1 = 10$.

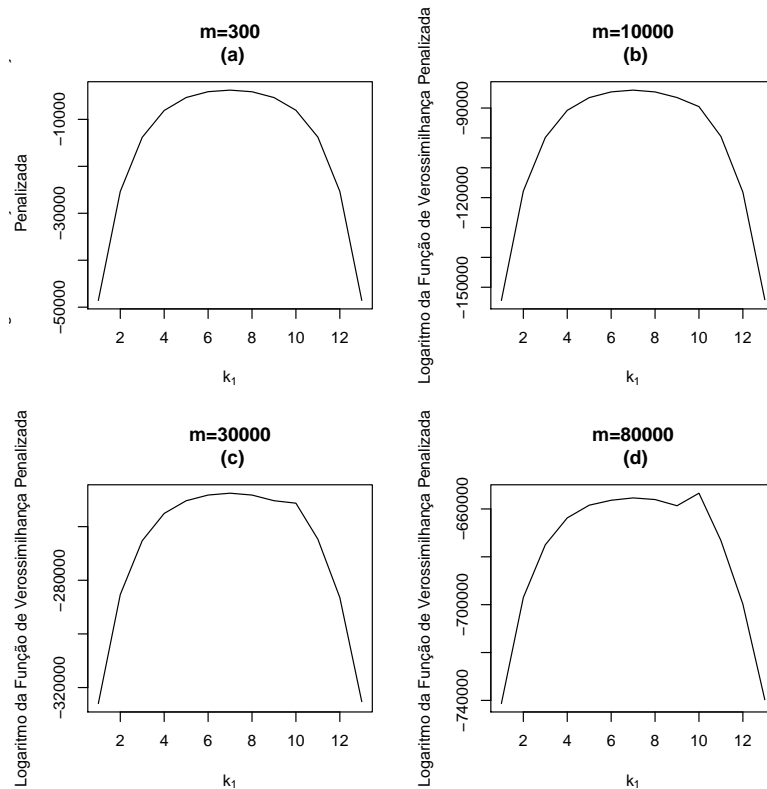


Figura 4.4: Gráfico do logaritmo da função de verossimilhança penalizada em relação ao ponto de corte em 2 segmentos, para diferentes valores de m .

Em todas as simulações apresentadas até o momento foi utilizado a constante $c = 1$ no algoritmo e vimos que, na demonstração da consistência do estimador $\hat{\mathbf{k}}_m$, a escolha de c não era relevante para m grande. Quando o tamanho da amostra é pequeno a escolha da constante c influencia no resultado final, principalmente quando n é grande. Dessa forma, propomos como constante o seguinte termo

$$c = \frac{1}{|\mathcal{A}|^{n\alpha}}, \quad 0 \leq \alpha \leq 1.$$

quando $\alpha = 0$ temos $c = 1$ e para $\alpha = 1$ a penalização será sempre um número entre 0 e 1, esse caso não nos interessa pois a verossimilhança penalizada será aproximadamente a verossimilhança usual. Assim vamos fazer $\alpha = \frac{1}{2}$ e aplicando nesta última simulação vemos que em vez de uma amostra $m=80000$ precisamos de uma com tamanho 25000 para encontrar o verdadeiro ponto de corte, ver Figura 4.5.

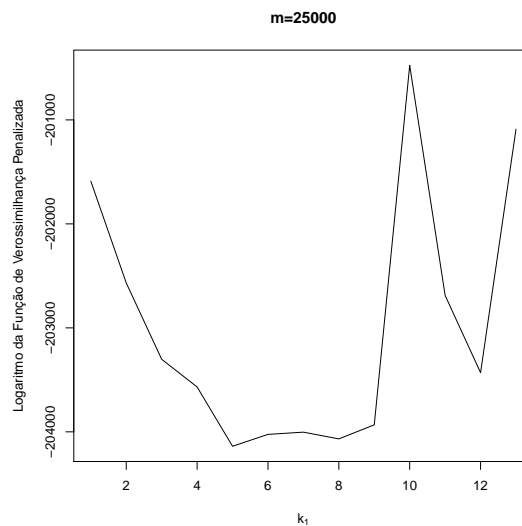


Figura 4.5: Gráfico do logaritmo da função de verossimilhança penalizada em relação ao ponto de corte em 2 segmentos quando $m = 25000$.

Apêndice A

Definições e Resultados Básicos

Definição A.1. Para duas distribuições \mathbb{P} e \mathbb{Q} em um conjunto finito \mathcal{B} a divergência de Kullback-Leibler ou divergência da informação é definida por

$$D(\mathbb{P}||\mathbb{Q}) = \sum_{b \in \mathcal{B}} \mathbb{P}(b) \log \left(\frac{\mathbb{P}(b)}{\mathbb{Q}(b)} \right). \quad (\text{A.1})$$

Essa divergência é definida se tanto \mathbb{P} quanto \mathbb{Q} somam 1 e se $\mathbb{Q}(b) = 0$ implica que $\mathbb{P}(b) = 0$ para todo b . Devemos interpretar a quantidade $0 \log 0$ como zero, caso ocorra, pelo fato que $\lim_{x \rightarrow 0} x \log x = 0$.

Proposição A.2. Sejam u_1, \dots, u_n e v_1, \dots, v_n números não negativos de forma que $u = \sum_{i=1}^n u_i$, $v = \sum_{i=1}^n v_i$, assim a desigualdade da soma dos logaritmos é dada por

$$\sum_{i=1}^n u_i \log \left(\frac{u_i}{v_i} \right) \geq u \log \left(\frac{u}{v} \right).$$

com igualdade se e somente se $\frac{u_i}{v_i}$ são iguais para todo i .

Demonstração. Seja

$$Z = \sum_{i=1}^n u_i \log \left(\frac{u_i}{v_i} \right) = \sum_{i=1}^n v_i \frac{u_i}{v_i} \log \left(\frac{u_i}{v_i} \right).$$

e considere f dada por $f(x) = x \log x$, em que $x = \frac{u_i}{v_i}$. Note que f é uma função convexa, logo

$$Z = \sum_{i=1}^n v_i \frac{u_i}{v_i} \log \left(\frac{u_i}{v_i} \right) = \sum_{i=1}^n v_i f \left(\frac{u_i}{v_i} \right).$$

fazendo $\sum_{i=1}^n v_i = v$ obtemos

$$Z = v \sum_{i=1}^n \frac{v_i}{v} f\left(\frac{u_i}{v_i}\right),$$

usando a desigualdade Jensen, temos

$$Z \geq v f\left(\sum_{i=1}^n \frac{v_i}{v} \frac{u_i}{v_i}\right) = v f\left(\frac{1}{v} \sum_{i=1}^n u_i\right)$$

usando o fato que $\sum_{i=1}^n u_i = u$, temos

$$Z \geq v f\left(\frac{u}{v}\right) = u \log\left(\frac{u}{v}\right).$$

a desigualdade permanece válida quando $n = \infty$ se e somente se $\sum_{i=1}^{\infty} v_i < \infty$ e $\sum_{i=1}^{\infty} u_i < \infty$.

Lema A.3. Para qualquer distribuições de probabilidade \mathbb{P} e \mathbb{Q} em \mathcal{B}

$$D(\mathbb{P}||\mathbb{Q}) \leq \sum_{b \in \mathcal{B}} \frac{(\mathbb{P}(b) - \mathbb{Q}(b))^2}{\mathbb{Q}(b)}$$

Demonstração. Usando a desigualdade $\log x \leq x - 1$ em (A.1) obtemos

$$D(\mathbb{P}||\mathbb{Q}) = \sum_{b \in \mathcal{B}} \mathbb{P}(b) \log\left(\frac{\mathbb{P}(b)}{\mathbb{Q}(b)}\right) \leq \sum_{b \in \mathcal{B}} \mathbb{P}(b) \left(\frac{\mathbb{P}(b)}{\mathbb{Q}(b)} - 1\right)$$

somando e subtraindo o termo $\sum_{b \in \mathcal{B}} \mathbb{Q}(b) \left(\frac{\mathbb{P}(b) - \mathbb{Q}(b)}{\mathbb{Q}(b)}\right)$ temos que

$$\begin{aligned} D(\mathbb{P}||\mathbb{Q}) &\leq \sum_{b \in \mathcal{B}} \left[\frac{(\mathbb{P}(b) - \mathbb{Q}(b))^2}{\mathbb{Q}(b)} \right] + \sum_{b \in \mathcal{B}} \mathbb{Q}(b) \left[\frac{(\mathbb{P}(b) - \mathbb{Q}(b))}{\mathbb{Q}(b)} \right] \\ &= \sum_{b \in \mathcal{B}} \left[\frac{(\mathbb{P}(b) - \mathbb{Q}(b))^2}{\mathbb{Q}(b)} \right] \end{aligned}$$

pois $\sum_{b \in \mathcal{B}} \mathbb{P}(b) = \sum_{b \in \mathcal{B}} \mathbb{Q}(b) = 1$, mais detalhes ver Csiszár & Talata (2006).

Proposição A.4. Para qualquer $\delta > 0$ e $b_{k_{j-1}+1}^{k_j} \in \mathcal{B}^{l_j}$ temos que

$$\left| \hat{\mathbb{P}}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}+1}^{k_j^{(r)}} \right) - \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}+1}^{k_j^{(r)}} \right) \right| < \sqrt{\frac{\delta \log m}{m}}$$

quase certamente, quando $m \rightarrow \infty$.

Demonstração. Vamos definir uma variável Z_i de forma que $\mathbb{E}(Z_i) = 0$. Assim considere

$$Z_i = I_{\left\{y_j^{(i)} = b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}}\right\}} - \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right), \quad i = 1, \dots, m.$$

A soma dessas m variáveis é

$$S_m = \sum_{i=1}^m Z_i = N_m^{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) - m \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right)$$

sendo que as variáveis Z_i são independentes e identicamente distribuídas. A esperança de Z_i é

$$\begin{aligned} \mathbb{E}(Z_i) &= \mathbb{E} \left(\hat{\mathbb{P}}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \right) - \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \\ &= 0 \end{aligned}$$

e a variância de Z_i é da forma

$$\begin{aligned} \text{Var}(Z_i) = \mathbb{E}(Z_i^2) &= \mathbb{E} \left(I_{\left\{y_j^{(i)} = b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}}\right\}} + \mathbb{P}_{\{k_{j-1}^*:k_j^*\}}^2 \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) - 2 I_{\left\{y_j^{(i)} = b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}}\right\}} \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \right) \\ &= \mathbb{E} \left(\hat{\mathbb{P}}_{\{k_{j-1}^*:k_j^*\}}^2 \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \right) - 2 \mathbb{P}_{\{k_{j-1}^*:k_j^*\}}^2 \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) + \mathbb{P}_{\{k_{j-1}^*:k_j^*\}}^2 \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \\ &= \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) - \mathbb{P}_{\{k_{j-1}^*:k_j^*\}}^2 \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \\ &= \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \left(1 - \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \right) \\ &\leq \frac{1}{4} \end{aligned}$$

Pela lei do Logaritmo Iterado (ver Billingsley (1995)), para algum $\epsilon > 0$, temos

$$|S_m| < \frac{(1 + \epsilon)}{4} \sqrt{2m \log \log m}$$

quase certamente quando $m \rightarrow \infty$. Em particular para $\epsilon < 3$

$$|S_m| < \sqrt{2m \log \log m}$$

$$\begin{aligned}
\frac{|S_m|}{m} &= \frac{\left| N_m^{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) - m \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \right|}{m} \\
&= \left| \hat{\mathbb{P}}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) - \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \right| \\
&< \frac{\sqrt{2m \log \log m}}{m} \\
&= \sqrt{\frac{2 \log \log m}{m}}
\end{aligned}$$

para qualquer $\delta > 0$ temos que

$$2 \log \log m < \delta \log m$$

logo

$$\left| \hat{\mathbb{P}}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) - \mathbb{P}_{\{k_{j-1}^*:k_j^*\}} \left(b_{k_{j-1}^{(r)}+1}^{k_j^{(r)}} \right) \right| < \sqrt{\frac{\delta \log m}{m}}.$$

quase certamente, quando $m \rightarrow \infty$.

Apêndice B

Rotinas em R

```
rm(list=ls(all=TRUE))

require(VLMC)
dad=c(1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 1, 2, 1, 1, 2, 1, 1, 2,
      2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2,
      1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, 1, 2, 2, 2, 1, 1, 1, 2,
      2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1)

mc=vlmc(dad) ## Segmento X2
draw(mc)
plot(as.dendrogram(mc))

transição<-function(a) ##a vetor de probabilidade de transição
{
u<-runif(1)
y<-ifelse(u<=a[1],1,2)
return(c(y,u))
}

dist.est=function(P){
P=matrix(P,ncol=2,nrow=2)
P0=diag(2)
for(i in 1:100){
P0=P0*%*%P
}
return(P0[1,])
}

simulação<-function(P1,n,k)
```

```

{
  pi1<-dist.est(P1)
  u<-runif(1)
  cadeia<-ifelse(u<=pi1[1],1,2)
  for(i in 2:(k))
  {
    cadeia[i]<-transição(P1[(cadeia[i-1]),1])
  }
  cadeia[(k+1):n]<-as.numeric(simulate.vlmc(mc,n-k)[1:(n-k)])

  return(cadeia)
}

P1=matrix(c(0.3,0.7,0.8,0.2),ncol=2,byrow=T) ### Matriz de Transição de X1
n=10;k=4;dados=NULL;m=2000;A=2

for(i in 1:m){
dados<-rbind(dados,simulação(P1,n,k))
}

c1=1 ### constante c

### 1 Segmento

l1=rep(0,m)
for(i in 1:(m)){
  l1[i]=do.call(paste, c(as.list(dados[i,1:n]), sep=""))
}

b1=sum(as.vector(table(l1)*log(table(l1)/m))) ## Logaritmo Verossimilhança

l1p=b1-c1*(A^n-1)*log(m) ## LVP

## 2 Segmentos

totid=matrix(0,nrow=m,ncol=(n-1))
for(i in 1:(m)){
for(j in 1:(n-1)){
  totid[i,j]=do.call(paste, c(as.list(dados[i,1:j]), sep=""))
}
}

```

```

totvo=matrix(0,nrow=m,ncol=(n-1))
for(i in 1:(m)){
for(j in (n-1):1){
  totvo[i,n-j]=do.call(paste, c(as.list(dados[i,(n-j+1):n]), sep=""))
}
}

tid=c()
tvo=c()
for(i in 1:(n-1)){
tid[i]=sum(as.vector(table(totid[,i])*log(table(totid[,i])/m)))
tvo[i]=sum(as.vector(table(totvo[,i])*log(table(totvo[,i])/m)))
}

l2p=rep(0,n-1)
for(i in 1:(n-1)){
l2p[i]=tid[i]+tvo[i]-c1*(A^(i)+A^(n-i)-2)*log(m)
}

l1p>max(l2p)

max(l2p)

plot(1:(n-1),l2p,type="l",ylab="Logaritmo da Função de Verossimilhança Penalizada",
main="1º Ponto de Corte",xlab=expression(k[1]))

k1=match(rep(max(l2p),length(l2p)),l2p)[1];k1

#####

## 3 Segmentos

## Verificar a existência de ponto de corte antes de k1

totid31=matrix(0,nrow=m,ncol=(k1-1))
for(i in 1:(m)){
for(j in 1:(k1-1)){
  totid31[i,j]=do.call(paste, c(as.list(dados[i,1:j]), sep=""))
}
}

```

```

totvo31=matrix(0,nrow=m,ncol=(k1-1))
for(i in 1:(m)){
for(j in (k1-1):1){
  totvo31[i,k1-j]=do.call(paste, c(as.list(dados[i,(k1-j+1):k1])), sep="")
}
}

tid31=c()
tvo31=c()
for(i in 1:(k1-1)){
tid31[i]=sum(as.vector(table(totid31[,i])*log(table(totid31[,i])/m)))
tvo31[i]=sum(as.vector(table(totvo31[,i])*log(table(totvo31[,i])/m)))
}

l31p=rep(0,k1-1)
for(i in 1:(k1-1)){
l31p[i]=tid31[i]+tvo31[i]-c1*(A^(i)+A^(k1-i)-2)*log(m)
}

l311=rep(0,m)
for(i in 1:(m)){
  l311[i]=do.call(paste, c(as.list(dados[i,1:(k1)])), sep="")
}

b31=sum(as.vector(table(l311)*log(table(l311)/m)))

l311p=b31-c1*(A^(k1)-1)*log(m);l311p

max(l311p)>max(l31p) ## Ver agora Segmento após o corte k1.

#####

## Ver se existe ponto de corte após k1

#####

totid32=matrix(0,nrow=m,ncol=(n-k1-1))
for(i in 1:(m)){
for(j in 1:(n-k1-1)){
  totid32[i,j]=do.call(paste, c(as.list(dados[i,(k1+1):(k1+j)])), sep="")
}
}

```

```

totvo32=matrix(0,nrow=m,ncol=(n-k1-1))
for(i in 1:(m)){
for(j in (n-k1-1):(1)){
  totvo32[i,n-k1-j]=do.call(paste, c(as.list(dados[i,(n-k1-j):(n-k1-1)]), sep=""))
}
}

tid32=c()
tvo32=c()
for(i in 1:(n-k1-1)){
tid32[i]=sum(as.vector(table(totid32[,i])*log(table(totid32[,i])/m)))
tvo32[i]=sum(as.vector(table(totvo32[,i])*log(table(totvo32[,i])/m)))
}

l32p=rep(0,n-k1-1)
for(i in 1:(n-k1-1)){
l32p[i]=tid32[i]+tvo32[i]-c1*(A^(i)+A^(n-k1-i)-2)*log(m)
}

l321=rep(0,m)
for(i in 1:(m)){
  l321[i]=do.call(paste, c(as.list(dados[i,(k1+1):(n)]), sep=""))
}

b32=sum(as.vector(table(l321)*log(table(l321)/m)))

l321p=b32-c1*(A^(n-k1)-1)*log(m);l321p

max(l321p)>max(l32p) ## Ver agora Segmento após o corte k1.

#####

#### Número correto de segmentos é 2 e o ponto de corte é k1=4

#####

###

rm(list=ls(all=TRUE))

require(VLMC)

```

```
dad=c(1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 1, 2, 1, 1, 2, 1, 1, 2,
      2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2,
      1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, 1, 2, 2, 2, 1, 1, 1, 2,
      2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1)
```

```
mc=vlmc(dad) ## Segmento X2
draw(mc)
```

```
dad1=c(2, 1, 2, 1, 2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 2, 1, 1, 1, 2, 1, 2, 1, 2, 1,
      2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 1,
      1, 2, 1, 1, 2, 1, 2, 1, 2, 1, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
      2, 2, 1, 2, 1, 1, 2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 1, 2, 1, 2, 1,
      2, 1, 2, 1, 2)
```

```
mc1=vlmc(dad1) ## Segmento X3
draw(mc1)
plot(as.dendrogram(mc1))
```

```
transição<-function(a) ## vetor de probabilidade de transição
{
  u<-runif(1)
  y<-ifelse(u<=a[1],1,2)
  return(c(y,u))
}
```

```
dist.est=function(P){
  P=matrix(P,ncol=2,nrow=2)
  P0=diag(2)
  for(i in 1:100){
    P0=P0%*%P
  }
  return(P0[1,])
}
```

```
simulação<-function(P1,n,kv1,kv2)
{
  pi1<-dist.est(P1)
  u<-runif(1)
  cadeia<-ifelse(u<=pi1[1],1,2)
  for(i in 2:(kv1))
  {
    cadeia[i]<-transição(P1[(cadeia[i-1]),1])
  }
}
```

```

    }
    cadeia[(kv1+1):kv2]<-as.numeric(simulate.vlmc(mc,kv2-kv1)[1:(kv2-kv1)])
    cadeia[(kv2+1):n]<-as.numeric(simulate.vlmc(mc1,n-kv2)[1:(n-kv2)])
    return(cadeia)
}

```

```

P1=matrix(c(0.3,0.7,0.8,0.2),ncol=2,byrow=T) ## Matriz de Transição de X1
n=12;kv1=4;kv2=9;dados=NULL;m=20000;A=2

```

```

for(i in 1:m){
dados<-rbind(dados,simulação(P1,n,kv1,kv2))
}

```

```

c1=1 ## constante c

```

```

#####

```

```

### 1 Segmento

```

```

l1=rep(0,m)
for(i in 1:(m)){
  l1[i]=do.call(paste, c(as.list(dados[i,1:n]), sep=""))
}

```

```

b1=sum(as.vector(table(l1)*log(table(l1)/m))) ## Logaritmo Verossimilhança

```

```

l1p=b1-c1*(A^n-1)*log(m) ## LVP

```

```

#####

```

```

##### 2 Segmentos

```

```

#####

```

```

totid=matrix(0,nrow=m,ncol=(n-1))
for(i in 1:(m)){
for(j in 1:(n-1)){
  totid[i,j]=do.call(paste, c(as.list(dados[i,1:j]), sep=""))
}
}

```

```

totvo=matrix(0,nrow=m,ncol=(n-1))
for(i in 1:(m)){

```

```

for(j in (n-1):1){
  totvo[i,n-j]=do.call(paste, c(as.list(dados[i,(n-j+1):n]), sep=""))
}
}

tid=c()
tvo=c()
for(i in 1:(n-1)){
  tid[i]=sum(as.vector(table(totid[,i])*log(table(totid[,i])/m)))
  tvo[i]=sum(as.vector(table(totvo[,i])*log(table(totvo[,i])/m)))
}

l2p=rep(0,n-1)
for(i in 1:(n-1)){
  l2p[i]=tid[i]+tvo[i]-c1*(A^(i)+A^(n-i)-2)*log(m)
}

l1p>max(l2p) ## Continua o processo

plot(1:(n-1),l2p,type="l",ylab="Logaritmo da Função de Verossimilhança Penalizada",
main="1° Ponto de Corte",xlab=expression(k[1]))

k1=match(rep(max(l2p),length(l2p)),l2p)[1];k1 ## Ponto de Corte

#####

## Ver se existe algum ponto de corte antes de k1

totid31=matrix(0,nrow=m,ncol=(k1-1))
for(i in 1:(m)){
  for(j in 1:(k1-1)){
    totid31[i,j]=do.call(paste, c(as.list(dados[i,1:j]), sep=""))
  }
}

totvo31=matrix(0,nrow=m,ncol=(k1-1))
for(i in 1:(m)){
  for(j in (k1-1):1){
    totvo31[i,k1-j]=do.call(paste, c(as.list(dados[i,(k1-j+1):k1]), sep=""))
  }
}

```

```

tid31=c()
tvo31=c()
for(i in 1:(k1-1)){
tid31[i]=sum(as.vector(table(totid31[,i])*log(table(totid31[,i])/m)))
tvo31[i]=sum(as.vector(table(totvo31[,i])*log(table(totvo31[,i])/m)))
}

l31p=rep(0,k1-1)
for(i in 1:(k1-1)){
l31p[i]=tid31[i]+tvo31[i]-c1*(A^(i)+A^(k1-i)-2)*log(m)
}

l311=rep(0,m)
for(i in 1:(m)){
  l311[i]=do.call(paste, c(as.list(dados[i,1:(k1)]), sep=""))
}

b31=sum(as.vector(table(l311)*log(table(l311)/m)))

l311p=b31-c1*(A^(k1)-1)*log(m);l311p

max(l311p)>max(l31p) ## Para esse segmento o processo é interrompido.

#####

## Ver se existe ponto de corte após k1

#####

totid32=matrix(0,nrow=m,ncol=(n-k1-1))
for(i in 1:(m)){
for(j in 1:(n-k1-1)){
  totid32[i,j]=do.call(paste, c(as.list(dados[i,(k1+1):(k1+j)]), sep=""))
}
}

totvo32=matrix(0,nrow=m,ncol=(n-k1-1))
for(i in 1:(m)){
for(j in (n-k1-1):(1)){
  totvo32[i,n-k1-j]=do.call(paste, c(as.list(dados[i,(n-k1-j):(n-k1-1)]), sep=""))
}
}

```

```

tid32=c()
tvo32=c()
for(i in 1:(n-k1-1)){
tid32[i]=sum(as.vector(table(totid32[,i])*log(table(totid32[,i])/m)))
tvo32[i]=sum(as.vector(table(totvo32[,i])*log(table(totvo32[,i])/m)))
}

l32p=rep(0,n-k1-1)
for(i in 1:(n-k1-1)){
l32p[i]=tid32[i]+tvo32[i]-c1*(A^(i)+A^(n-k1-i)-2)*log(m)
}

l321=rep(0,m)
for(i in 1:(m)){
  l321[i]=do.call(paste, c(as.list(dados[i,(k1+1):(n)]), sep=""))
}

b32=sum(as.vector(table(l321)*log(table(l321)/m)))

l321p=b32-c1*(A^(n-k1)-1)*log(m);l321p

max(l321p)>max(l32p) ## Continua o processo somente para esse segmento

k2=k1+match(rep(max(l32p),length(l32p)),l32p)[1];k2

### Temos que analisar se existe mais algum ponto de corte

plot((k1+1):(n-1),l32p,type="l",ylab="Logaritmo da Função de Verossimilhança Penalizada",
main="2º Ponto de Corte",xlab=expression(k[2]))

#####

#### Verificar se existe ponto de corte entre k1 e k2

totid41=matrix(0,nrow=m,ncol=(k2-k1-1))
for(i in 1:(m)){
for(j in 1:(k2-k1-1)){
  totid41[i,j]=do.call(paste, c(as.list(dados[i,(k1+1):(k1+j)]), sep=""))
}
}

```

```

totvo41=matrix(0,nrow=m,ncol=(k2-k1-1))
for(i in 1:(m)){
for(j in (k2-k1-1):1){
  totvo41[i,k2-k1-j]=do.call(paste, c(as.list(dados[i,(k2-k1-j):(k2-k1-1)]), sep=""))
}
}

tid41=c()
tvo41=c()
for(i in 1:(k2-k1-1)){
tid41[i]=sum(as.vector(table(totid41[,i])*log(table(totid41[,i])/m)))
tvo41[i]=sum(as.vector(table(totvo41[,i])*log(table(totvo41[,i])/m)))
}

l41p=rep(0,k2-k1-1)
for(i in 1:(k2-k1-1)){
l41p[i]=tid41[i]+tvo41[i]-c1*(A^(i)+A^(k2-k1-i)-2)*log(m)
}

## Ver Segmento inteiro

l41=rep(0,m)
for(i in 1:(m)){
  l41[i]=do.call(paste, c(as.list(dados[i,(k1+1):(k2)]), sep=""))
}

b41=sum(as.vector(table(l41)*log(table(l41)/m)))

l14p=b41-c1*(A^(k2-k1)-1)*log(m)

max(l14p)>max(l41p) ## Analisar outro Segmento

#####

### Verificar a existência de ponto de corte após k2

totid42=matrix(0,nrow=m,ncol=(n-k2-1))
for(i in 1:(m)){
for(j in 1:(n-k2-1)){
  totid42[i,j]=do.call(paste, c(as.list(dados[i,(k2+1):(k2+j)]), sep=""))
}
}

```

```

totvo42=matrix(0,nrow=m,ncol=(n-k2-1))
for(i in 1:(m)){
for(j in (n-k2-1):(1)){
  totvo42[i,n-k2-j]=do.call(paste, c(as.list(dados[i,(n-k2-j):(n-k2-1)]), sep=""))
}
}

tid42=c()
tvo42=c()
for(i in 1:(n-k2-1)){
tid42[i]=sum(as.vector(table(totid42[,i])*log(table(totid42[,i])/m)))
tvo42[i]=sum(as.vector(table(totvo42[,i])*log(table(totvo42[,i])/m)))
}

l42p=rep(0,n-k2-1)
for(i in 1:(n-k2-1)){
l42p[i]=tid42[i]+tvo42[i]-c1*(A^(i)+A^(n-k2-i)-2)*log(m)
}

### Analisar Segmento Inteiro

l42=rep(0,m)
for(i in 1:(m)){
  l42[i]=do.call(paste, c(as.list(dados[i,(k2+1):n]), sep=""))
}

b42=sum(as.vector(table(l42)*log(table(l42)/m)))

l24p=b42-c1*(A^(n-k2)-1)*log(m)

max(l24p)>max(l42p) ### Fim

#####

#### Temos 3 Segmentos e os pontos de corte são k1=4 e k2=9

```


Referências Bibliográficas

- [1] Bernaola-Galván, P., Román-Roldán, R. & Oliver, J. L. (1996). Compositional segmentation and long-range fractal correlations in DNA sequences. *Physical Review E* **53**(5): 5181-5189.
- [2] Billingsley, P. (1995). *Probability and Measure*. 3 ed. Wiley, New York.
- [3] Boys, R. J., Henderson, D. A. & Wilkinson, D.J. (2000). Detecting homogeneous segments in DNA sequences by using hidden Markov models. *Applied Statistics* **49**, 269-285.
- [4] Boys, R. J. & Henderson, D. A. (2004). A bayesian approach to DNA sequence segmentation. *Biometrics* **60**, 573-588.
- [5] Braun, J. V. & Muller, H. -G. (1998). Statistical methods for DNA sequence segmentation. *Statistical Science* **13**(2), 142-162.
- [6] Bühlmann, P. & Wyner, A. J. (1999). Variable length Markov chains. *Annals of Statistics* **27**, 480-513.
- [7] Csiszár, I. & Talata, Z. (2006). Context tree estimation for not necessarily finite memory process, via bic and mdl. *IEEE Transactions on Information Theory*. **52**(3), 1007-1016.
- [8] Gwadera, R., Gionis, A. & Mannila, H. (2008). Optimal segmentation using tree models. *Knowledge and Information Systems*. **15**(3), 259-283.
- [9] Li, W. (2001a). Dna segmentation as a model selection process. *RECOMB*: 204-210.
- [10] Li, W. (2001b). New stopping criteria for segmentation DNA sequences. *Physical Review Letters* **86**: 5815-5818.
- [11] Mächler, M. & Bühlmann, P. (2004). Variable length Markov chains: methodology, computing and software. *Journal of Computational and Graphical Statistics* **13**: 435-455.
- [12] Nur, D., Allingham, D., Rousseau, J., Mengersen, K. L. & McVinish, R. (2009). Bayesian hidden Markov models for DNA sequence segmentation: A prior sensitivity analysis. *Computational Statistics and Data Analysis* **53**, 1873-1882.

-
- [13] Oliver, J.L., Román-Roldán, R., & Bernaola-Galván, P. (1999). SEGMENT: identifying compositional domains in DNA sequence. *Bioinformatics* **15**(12): 974-979.
- [14] Rissanen, J. (1983). A universal data compression system. *IEEE Transactions on Information Theory* **29**(5), 656-664.
- [15] Terzi, E. & Tsaparas, P. (2006). Efficient algorithms for sequence segmentation. *Proceeding of SIAM International Conference on Data Mining*, 314-325.
- [16] Willems, F., Shtarkov, Y. & Tjalkens, T. (2000). Context tree maximizing. *Conference on Information Sciences and Systems*, 7-12.