



Introdução à Teoria da Informação

Prof. Leonardo Vidal Batista
DSC/CI/UFPB, PPGI/UFPB, PPGCC/UFCG
leonardo@ci.ufpb.br

Teoria da Informação

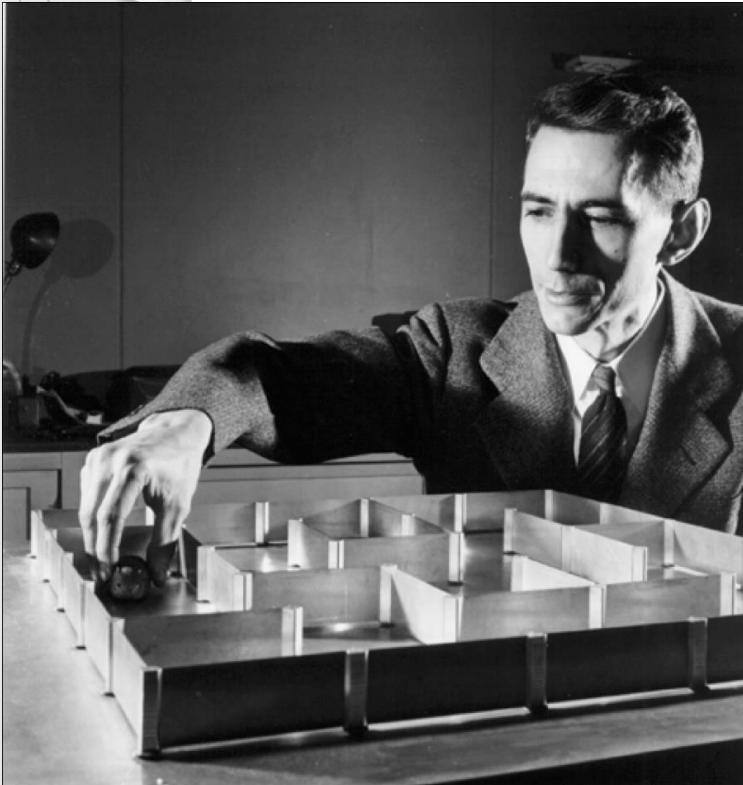
Claude Shannon, 1948, “A Mathematical Theory of Communications”

Área que estuda os processos de comunicação

Conceituação formal de *informação* e da *capacidade de transmissão de informação* pelos canais de comunicação

Compressão de Dados: aplicação da Teoria da Informação

Claude Shannon



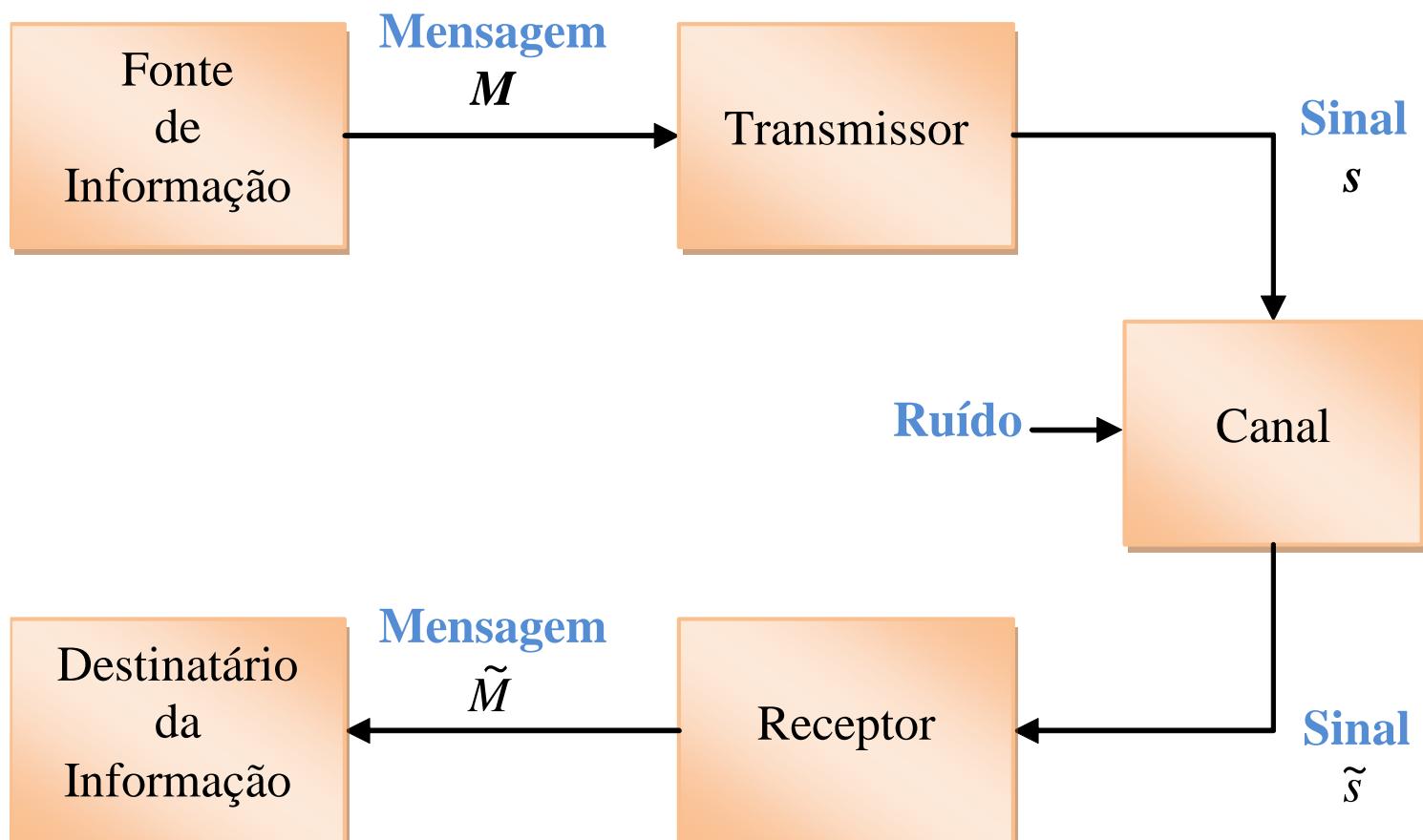
1916-2001

Pai da Era da Informação

Lógica booleana para
otimização de circuitos elétricos

Pioneiro de IA, robótica,
criptografia...

Sistemas de comunicação



Questões

Pode-se garantir comunicação perfeita?

Pode-se garantir comunicação confiável?

Comunicação equivale a "transmissão de dados"?

Respostas de Shannon:

Não!

Sim. Pode-se construir sistemas de comunicação tão confiáveis (estatisticamente) quanto se deseje: 1 bit errado em 1 bilhão, 1 em 1 trilhão...

Não. Comunicação envolve transmissão de informação.

Compressores sem perdas reduzem a quantidade de dados em uma mensagem, preservando a informação

O Que É “Informação”?

Informação (ou entropia) é uma medida associada à incerteza ou surpresa relacionada a um evento.

O processo de comunicação é inherentemente probabilístico.

Informação é relativa!

Informação e Incerteza

Para a próxima aula, decorar letra por letra o livro “Grande Sertão: Veredas”.

Na aula, farei apenas a leitura do livro.

Transmitirei informação?

Ocorrerá comunicação?

Teoria da Informação e Física

A equação de entropia (informação) de Shannon é análoga à entropia da termodinâmica

Era Newton: matéria, energia e informação

Era Einstein: energia e informação

Era Shannon: informação

Relatividade da Informação

010101010101...

Pedro Álvares Ca...

Marcela amou-me durante quinze meses e
onze ...

14159265...

27033903618375114...

Entropia e Compressão

A entropia é um limite para o grau de compressão máxima que se pode obter

Entropia elevada \Rightarrow grau de compressão reduzido

Entropia reduzida \Rightarrow grau de compressão elevado

Procurar construir modelos que reduzam a entropia!

Modelagem e Compressão

Um bom modelo prevê com precisão símbolos sucessivos da mensagem

Bom modelo \Rightarrow surpresa reduzida \Rightarrow entropia reduzida \Rightarrow grau de compressão elevado

Grau de compressão elevado \Rightarrow entropia reduzida \Rightarrow surpresa reduzida \Rightarrow bom modelo

Bom modelo \Leftrightarrow grau de compressão elevado

Modelagem e Compressão

Grau de compressão é uma medida da qualidade de um modelo

Entropia da Língua Inglesa

Qual é a entropia da Língua Inglesa?

O que é “língua inglesa”???

Qual é a entropia da língua inglesa nos livros de Swift?

Entropia da Língua Inglesa

Qual é a entropia da língua inglesa nos livros de Swift?

Por um especialista em Literatura Inglesa?

Por um engenheiro inglês?

Por um modelo computacional?

Por uma criança recém-alfabetizada em inglês?

Entropia da Língua Inglesa

Shannon, 1952: entropia da "língua inglesa" escrita, percebida por americanos adultos ≈ 1 bit/símbolo

Estudos posteriores confirmaram a estimativa

Entropia da Língua Inglesa

Melhores modelos computacionais: 1,2 bit/símbolo

Quando um modelo computacional tornar-se tão bom quanto um humano, o computador terá "entendido" os textos?

Entropia da Língua Inglesa

Aplicações para um bom modelo computacional de um idioma: OCR; conversão de voz para texto; síntese de voz; atribuição de autoria etc

Dificuldades: Pedro Álvares Ca...

Entropia de outras mensagens

Língua portuguesa

DNA

IBOVESPA

Dados sísmicos

Áudio

Imagens

Vídeo

etc

Compressão para não Comprimir!

Compressores modernos possuem excelente capacidade de modelagem estatística

Os modelos podem ser utilizados para outros fins que não compressão

- Classificação de microorganismos

- Segmentação e classificação de texturas

- Atribuição de autoria

Grau de compressão: qualidade do modelo!

Compressão para Comprimir

Tecnologia-chave em

DVDs

TV digital

MP3

Câmeras fotográficas e de vídeo

Redes de comunicação

Etc

Sugestões de Projetos

Redução de artefatos JPEG utilizando autoencoders

Estimativa de entropia da língua portuguesa usando um compressor ótimo e ajuste de curvas

Jogo de Shannon com ChatGPT

Classificação de textos por período literário utilizando LZW e PPM

Classificação de sentimentos (texto)

- Avaliação de filmes

- Avaliação de produtos

Classificação de discurso de ódio ou ofensivo

Classificação de spam

Classificação de vieses em texto

<https://paperswithcode.com/datasets?task=text-classification>

<https://imerit.net/blog/17-best-text-classification-datasets-for-machine-learning-all-pbm/>

Jogo de Shannon com ChatGPT

Me: You are about to play the Shannon Game. This game involves predicting the next character in a 15-letter sentence. The possible characters include all the letters of the English alphabet, punctuation marks, and spaces. In the sentences I will present to you, spaces are represented by an underscore character.

ChatGPT: Sounds like a fun challenge! Please go ahead and give me the 15-letter sentence, and I'll try to predict the next character.

Me: police_vehicle_

ChatGPT: For the sentence "police_vehicle_", I predict the next character might be 'p' if it's referring to a specific type or model, like "police_vehicle_park" for a place where police vehicles are stationed. Another possibility could be 'i' if it refers to a specification or feature, like "police_vehicle_identification".

Me: You don't have to explain your rationale. Just give me a list of 10 possible characters, in decreasing order of probability.

ChatGPT: Sure, here are 10 possible characters, in decreasing order of probability: 'i' 's' 'p' 'r' 'c' 'm' 't' 'a' 'd' 'o'

Me: On_the_way,_the

ChatGPT: 'y' 'r' 'm' 'c' 's' 't' 'b' 'p' 'l' 'n'

Me: Why haven't you considered the possibility of the space (underscore) being next?

ChatGPT: You're right to point that out. I should have considered the space (underscore) as a likely next character, especially since it often appears in phrases to separate words. Let me provide an updated list with the underscore included, considering its relevance in separating words in English sentences: '_ 'y' 'r' 'm' 'c' 's' 't' 'b' 'p' 'l'

Fonte de Informação

Fonte de informação: elemento de um processo de comunicação que produz informação

Fonte discreta de informação, S : produz *mensagens*, que são seqüências de elementos selecionados de um conjunto $A = \{a_0, a_1, a_2, \dots\}$ (*alfabeto da fonte*)

a_i : *símbolo ou letra*

Exemplos: fontes decimais e binárias

Fonte de dígitos decimais

$$A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

Mensagem M = 590126885122380546

Fonte binária:

$$A = \{0, 1\}$$

Representação da Informação

Semáforo de três cores: informação de interesse são cores

Representação da informação

Texto em inglês: {R, Y, G}; {Red, Yellow, Green}

Texto em português: {Vermelho, Amarelo, Verde}

Computador: {00, 01, 10}

Outras aplicações: {0, 1, 2}; cores; sons; voltagens

Codificação

Representação da informação

Palavra-código: representação de um símbolo ou sequência de símbolos

Código de fonte: conjunto de palavras-código capaz de representar qualquer mensagem emitida pela fonte

Codificação

Alguns códigos para fonte de dígitos decimais

Algarismos arábicos	ASCII	Binário 1	Binário 2	Binário 3	Octal	Alfabético
0	30_{16}	0000_2	00_2	111111_2	00_8	a
1	31_{16}	0001_2	01_2	111110_2	01_8	f
2	32_{16}	0010_2	10_2	111101_2	02_8	k
3	33_{16}	0011_2	1100_2	111100_2	03_8	m
4	34_{16}	0100_2	1101_2	1110_2	04_8	n
5	35_{16}	0101_2	1110_2	1101_2	05_8	b
6	36_{16}	0110_2	111100_2	1100_2	06_8	z
7	37_{16}	0111_2	111101_2	10_2	07_8	x
8	38_{16}	1000_2	111110_2	01_2	10_8	p
9	39_{16}	1001_2	111111_2	00_2	11_8	s

Codificador e Decodificador

Codificador: elemento (ser humano, circuito, programa etc) que representa as mensagens geradas pela fonte empregando um código específico

Decodificador: elemento responsável por restaurar as mensagens originais a partir da mensagem codificada

Códigos de Propósito Específico

Códigos de transmissão: Morse,
códigos corretores de erros etc

Códigos de criptografia

Códigos de compressão: representação
reduzida

- Códigos de compressão binários: menor
número de bits possível

Códigos de Compressão

Em termos de compressão, qual é o melhor código binário?

Decimal	Binário 1	Binário 2	Binário 3
0	0000_2	00_2	111111_2
1	0001_2	01_2	111110_2
2	0010_2	10_2	111101_2
3	0011_2	1100_2	111100_2
4	0100_2	1101_2	1110_2
5	0101_2	1110_2	1101_2
6	0110_2	111100_2	1100_2
7	0111_2	111101_2	10_2
8	1000_2	111110_2	01_2
9	1001_2	111111_2	00_2

Códigos de Compressão

Depende da frequência relativa
(probabilidade) com que a fonte produz
cada símbolo.

Exemplo1: codificação-decodificação

Codificar e decodificar a mensagem
RGBRBRR usando RGB1, RGB2, RGB3

Símbolo	Código RGB1	Código RGB2	Código RGB3
Cor vermelha	00	0	0
Cor verde	01	1	10
Cor azul	10	00	11

RGB1

Codificação: 00011000100000 (14 bits)

Decodificação: RGBRBRR

Exemplo1 (cont.)

Codificar e decodificar a mensagem
RGBRBRR usando RGB1, RGB2, RGB3

Símbolo	Código RGB1	Código RGB2	Código RGB3
Cor vermelha	00	0	0
Cor verde	01	1	10
Cor azul	10	00	11

RGB2

Codificação: 010000000 (9 bits)

Decodificação: RG ?

Exemplo 1 (cont.)

Codificar e decodificar a mensagem
RGBRBRR usando RGB1, RGB2, RGB3

Símbolo	Código RGB1	Código RGB2	Código RGB3
Cor vermelha	00	0	0
Cor verde	01	1	10
Cor azul	10	00	11

RGB3

Codificação: 0101101100 (10 bits)

Decodificação: RGBRBRR

Códigos de Prefixo

RGB1 e RGB3 são códigos de prefixo:
nenhuma palavra-código é prefixo de
outra

Códigos unicamente decodificáveis

RGB2 não é um código de prefixo

Exemplo 2: comprimento médio

Codificar mensagem W usando Bin2

$W = 102450010211000703060090002450$
 $233004050060078600007688000785$

Alfabeto: {0, 1, ..., 9}

Comprimento da mensagem codificada, em bits?

Comprimento médio, em bits/símbolo?

Decimal	Bin2
0	00_2
1	01_2
2	10_2
3	1100_2
4	1101_2
5	1110_2
6	111100_2
7	111101_2
8	111110_2
9	111111_2

Exemplo 2: comprimento médio

$N = 60$ símbolos

N_i : Nº de ocorrências de a_i

$N_0=29; N_1=4; N_2=4;$

$N_3=3; N_4=3; N_5=4;$

$N_6=4; N_7=4; N_8=4; N_9=1$

b_i : Nº de bits da palavra-código associada a a_i

$b_0=b_1=b_2=2; b_3=b_4=b_5=4;$

$b_6=b_7=b_8=b_9=6$

Decimal	Bin2
0	00_2
1	01_2
2	10_2
3	1100_2
4	1101_2
5	1110_2
6	111100_2
7	111101_2
8	111110_2
9	111111_2

Exemplo 2: comprimento médio

Total de bits: $l = b_0N_0 + b_1N_1 + \dots + b_9N_9$

Número médio de bits por símbolo: $\bar{l} = l/N$

$$\begin{aligned} l &= 29 \times 2 + 4 \times 2 + 4 \times 2 + 3 \times 4 + 3 \times 4 + 4 \times 4 + \\ &\quad + 4 \times 6 + 4 \times 6 + 4 \times 6 + 1 \times 6 = 192 \text{ bits} \end{aligned}$$

$$\begin{aligned} \bar{l} &= (29 \times 2 + 4 \times 2 + 4 \times 2 + 3 \times 4 + 3 \times 4 + 4 \times 4 + \\ &\quad + 4 \times 6 + 4 \times 6 + 4 \times 6 + 1 \times 6) / 60 \\ &= (29/60) \times 2 + (4/60) \times 2 + (4/60) \times 2 + \\ &\quad + (3/60) \times 4 + (3/60) \times 4 + (4/60) \times 4 + \\ &\quad (4/60) \times 6 + (4/60) \times 6 + (4/60) \times 6 + (1/60) \times 6 \\ &= 3,2 \text{ bits/símbolo.} \end{aligned}$$

Exemplo 2: comprimento médio

Para alfabeto $A = \{a_0, a_1, \dots, a_{M-1}\}$

P_i : estimativa da probabilidade de a_i

P_i = freqüência relativa de a_i

$$P_i = N_i/N$$

$$\bar{l} = P_0 b_0 + P_1 b_1 + \dots + P_{M-1} b_{M-1}$$

$$\bar{l} = \sum_{i=0}^{M-1} P_i b_i$$

Exemplo 3: comprimento médio

Codificar a mensagem W usando Bin3

$$\begin{aligned}b_0 &= b_1 = b_2 = b_3 = 6; \\b_4 &= b_5 = b_6 = 4; \quad b_7 = b_8 = b_9 = 2\end{aligned}$$

$$\begin{aligned}\bar{l} &= 29 \times 6 + 4 \times 6 + 4 \times 6 + 3 \times 6 + 3 \times 4 + \\&+ 4 \times 4 + 4 \times 4 + 4 \times 2 + 4 \times 2 + 2 \times 1) / 60 = \\&= 5,03 \text{ bits/símbolo}\end{aligned}$$

Para compressão:

Bin2 é melhor que Bin3

Há algum código de prefixo melhor?

Bin3
111111_2
111110_2
111101_2
111100_2
1110_2
1101_2
1100_2
10_2
01_2
00_2

Problema Essencial da Compressão

Dado um modelo probabilístico para a fonte, encontrar um código unicamente decodificável ótimo, ou seja, que represente a saída da fonte com o menor número de bits possível

Encontrar b_i de forma a minimizar (com decodificabilidade)

$$\bar{l} = \sum_{i=0}^{M-1} P_i b_i$$

Informação

Solução: $b_i = I_i = \log \frac{1}{P_i}$

b_i = Informação I_i associada ao símbolo a_i :

Se \log_2 : informação em bits

Se \log_3 : informação em trits

Se \ln : informação em nats

Se \log_{10} : informação em dígitos decimais

Entropia

$$H = \bar{l}_{\min} = \sum_{i=0}^{M-1} P_i \log_2 \frac{1}{P_i}$$

Unidade (para base 2): bits/símbolo

Entropia da fonte ou da mensagem?

Shannon, 1948: entropia e compressão

Huffman, 1952: codificação ótima de comprimento inteiro

Pesquisadores da IBM, 1978: codificação ótima (codificação aritmética)

Exemplo 4: Informação e entropia

W = 1 0 2 4 5 0 0 1 0 2 1 1 0 0 0 7 0 3 0 6
0 0 9 0 0 0 2 4 5 0 2 3 3 0 0 4 0 5 0 0
6 0 0 7 8 6 0 0 0 0 7 6 8 8 0 0 0 7 8 5

$$P_0 = 29/60 \Rightarrow I_0 = \log_2 60/29 = 1,05 \text{ bit}$$

$$P_1 = P_2 = P_5 = P_6 = P_7 = P_8 = 4/60 \\ \Rightarrow I_1 = I_2 = I_5 = I_6 = I_7 = I_8 = \log_2 60/4 = 3,91 \text{ bits}$$

$$P_3 = P_4 = 3/60 \Rightarrow I_3 = I_4 = \log_2 60/3 = 4,32 \text{ bits} \\ P_9 = 1/60 \Rightarrow I_9 = \log_2 60 = 5,91 \text{ bits}$$

$$H = (29/60) \times 1,05 + (4/60) \times 3,91 \\ + \dots + (1/60) \times 5,91 = 2,602 \text{ bits/símbolo}$$

Exemplo 4: Informação e entropia

Código eficaz?

Comparar com a entropia!

Comparar codificação por Bin2 e Bin3
(3,2 bits/símbolo e 5,03 bits/símbolo)
com $H = 2,602$ bits/símbolo

Código de Huffman

Código de comprimento inteiro ótimo para a mensagem dos exemplos anteriores:

Símbolo	Código de Huffman	Comprimento Palavra-código	Informação
0	1	1	1,05
1	0111	4	3,91
2	0110	4	3,91
5	0101	4	3,91
6	0100	4	3,91
7	0011	4	3,91
8	0010	4	3,91
3	0001	4	4,32
4	00001	5	4,32
9	00000	5	5,91

Código de Huffman

$$l_{Huff} = 29 \times 1 + 4 \times 4 + 4 \times 4 + \dots + 1 \times 5 = 157 \text{ bits}$$

$$\bar{l}_{Huff} = l_{Huff} / 60 = 2,62 \text{ bits/símbolo.}$$

Entropia $H = 2,60$

Eficácia?

Código de comprimento fixo de 4 bits/símbolo: $RC = 60 \times 4 / (60 \times 2,62) = 1,53$
(1,53:1)

Código de Huffman

Outras árvores de Huffman?

Como quebrar empates?

Que informação passar ao decodificador para permitir decodificação?

Tabela de códigos ou

Árvore ou

Contadores

Quando o Huffman se torna ineficaz?

Código de Huffman

Redundância do código

$$R = \bar{l} - H$$

$$R_p = \frac{(\bar{l} - H)}{H} \times 100$$

Pode-se mostrar que

$$R_{Huff} \leq 1 \Leftrightarrow \bar{l}_{Huff} \leq H + 1$$

Quando o Huffman se torna ineficaz?

Quando $P_i >> 0,5$

Código de Huffman - Exemplo

$A = \{a, b, c, d, r\}$

$P(a) = 0,99; P(b)=P(c)=P(d)=0,003, P(r) = 0,001$

$I(a) = 0,0145; I(b)=I(c)=I(d)= 8,3817; I(r)=9,9668$

Comprimento médio de Huffman, entropia, eficácia, RC?

$$\bar{l}_{Huff} = 0,99 \times 1 + 3 \times 0,003 \times 3 + 0,001 \times 3 = \\ 1,018 \text{ bit/símb}$$

$$RC = 3/1,018 = 2,947:1$$

$$H = 0,0998 \text{ bits/símbolo.}$$

Código de Huffman - Exemplo

$$R_{Huff} = 0,9202 \text{ bit/símbol o}$$

$$R_p = 922\%$$

$$\text{RC}_{\max} = 3/0,0998 = 30,06:1$$

$$\text{RC}_{\max} = 10,22 \times \text{RC}_{\text{Huff}}$$

O que fazer para aplicar o código de Huffman eficientemente nestes casos?

RLE, por exemplo

Huffman para alfabetos binários?

Modelos Probabilísticos

Programação =
estrutura de dados + algoritmos

Compressão =
modelagem + codificação

Modelos Probabilísticos

Equiprobabilidade, $|A|$: entropia?

Equiprobabilidade, $|A| = 2^k$: entropia?

Equiprobabilidade \Leftrightarrow Entropia máxima

$$H = \log_2 |A|$$

$$\text{Se } |A| = 2^k, H = k$$

$P(a_i) = 1$: entropia?

$P(a_i) = 1 \Leftrightarrow$ entropia mínima, $H = 0$

Modelos Probabilísticos

Estáticos: probabilidades estáticas durante todo o processo de codificação

Semiadaptativos: probabilidades estimadas para cada mensagem a ser codificada, e mantidas estáticas durante a codificação

Adaptativos: probabilidades atualizadas durante a codificação

Modelos Probabilísticos

Não-contextuais: símbolos independentes, probabilidades não condicionais.

Contextuais: símbolos dependentes, probabilidades condicionais.

Modelos contextuais estáticos, não-contextuais semi-adaptativos, etc.

Modelos Probabilísticos

Modelagem + Codificação: Huffman
não-contextual semi-adaptativo,
aritmético contextual adaptativo etc

Família LZ (contextuais adaptativos)

Família PPM (contextuais adaptativos)

Equivalência entre compressores
baseados em dicionários e
compressores estatísticos

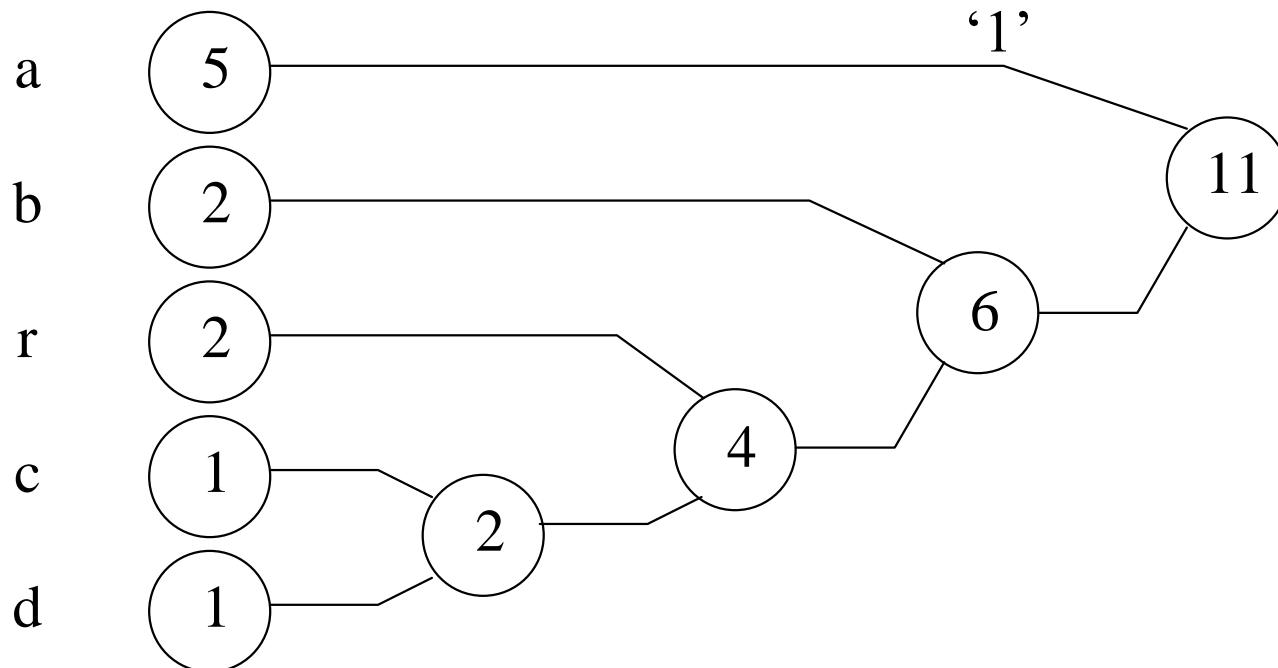
Exemplo: Huffman Adaptativo

Huffman não contextual adaptativo de decremento

W = abracadabra

Inicia-se como no Huffman não-contextual semiadaptativo visto em aula.

Exemplo: Huffman Adaptativo



‘a’ codificado com palavra-código ‘1’

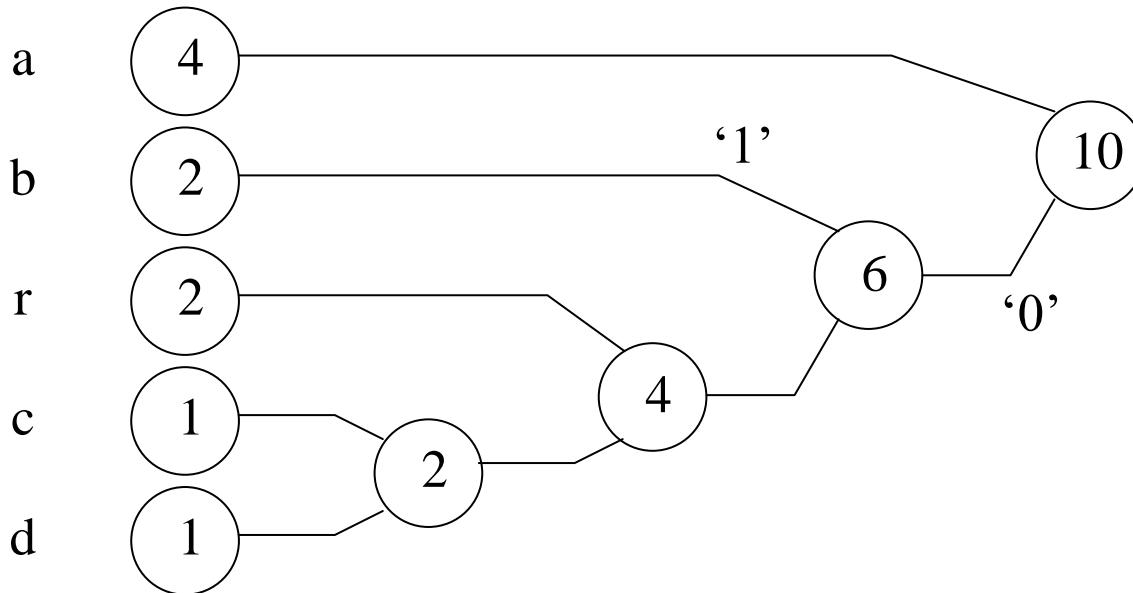
Exemplo: Huffman Adaptativo

$$I_0(a) = \log_2 (11/5) = 1,14 \text{ bit}$$

As estimativas de probabilidades são alteradas para refletir a ocorrência do símbolo ‘a’

Atualização do modelo e reconstrução da árvore para codificação do próximo símbolo:

Exemplo: Huffman Adaptativo

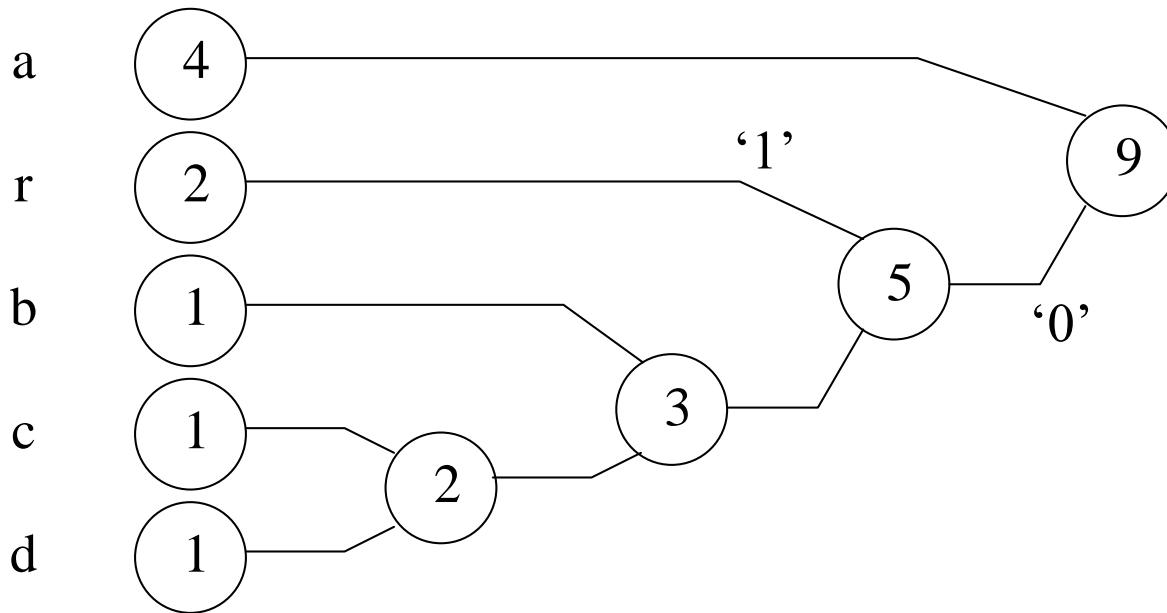


'b' codificado com palavra-código '01'

$$I_1(b) = \log_2 (10/2) = 2,32 \text{ bits.}$$

Probabilidades alteradas pela ocorrência de 'b':

Exemplo: Huffman Adaptativo

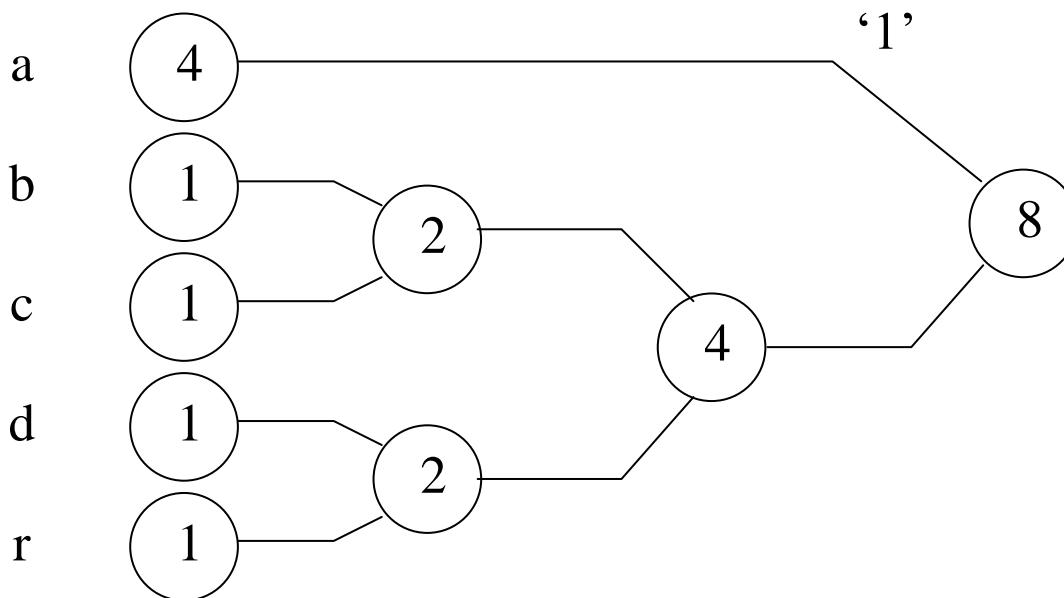


'r' codificado com '01' (= código de 'b' no passo anterior!!!)

$$I_2(r) = \log_2 (9/2) = 2,17 \text{ bits.}$$

Probabilidades e árvore alteradas

Exemplo: Huffman Adaptativo

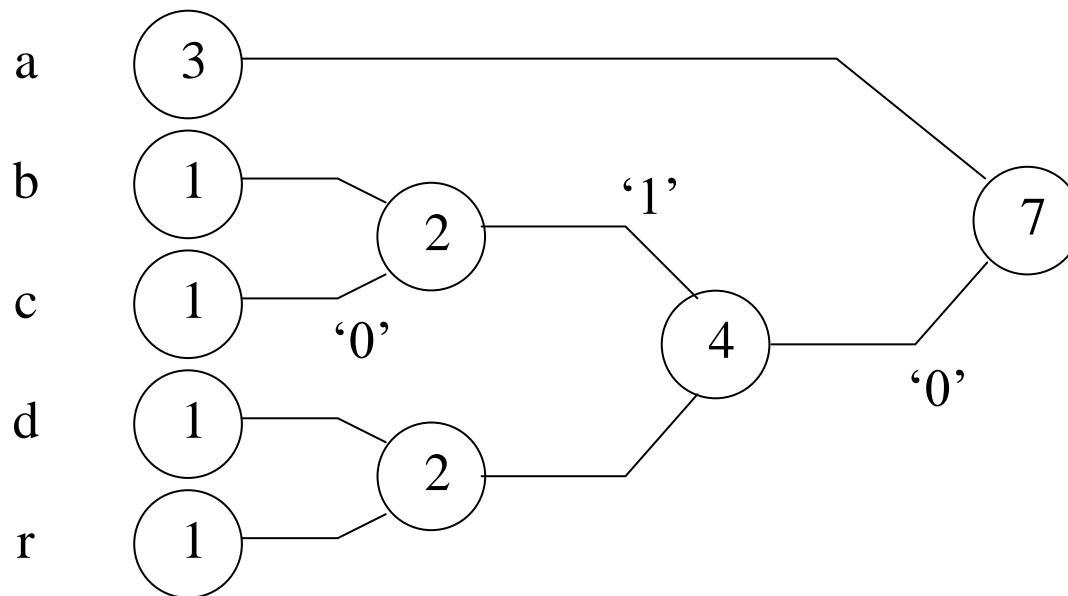


'a' codificado com '1'

$$I_3(a) = \log_2 (8/4) = 1 \text{ bit.}$$

Probabilidades e árvore alteradas

Exemplo: Huffman Adaptativo

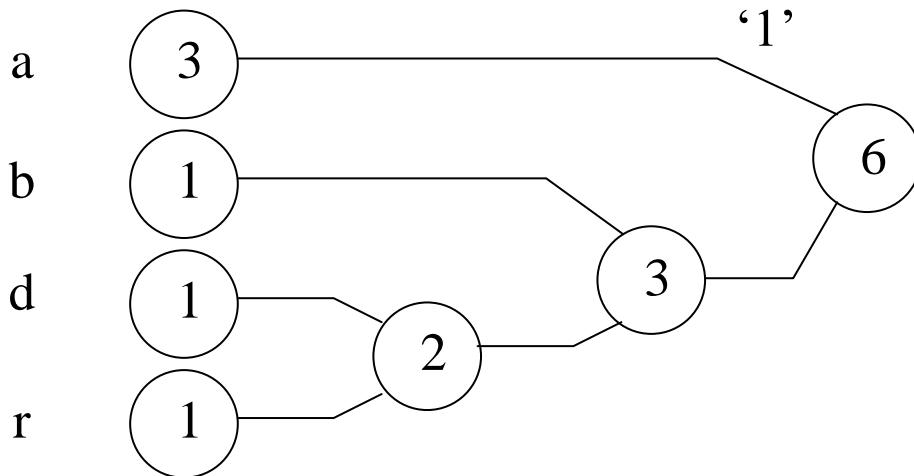


'c' codificado com '010'

$$I_4(c) = \log_2 (7/1) = 2,8 \text{ bit.}$$

Probabilidades e árvore alteradas

Exemplo: Huffman Adaptativo

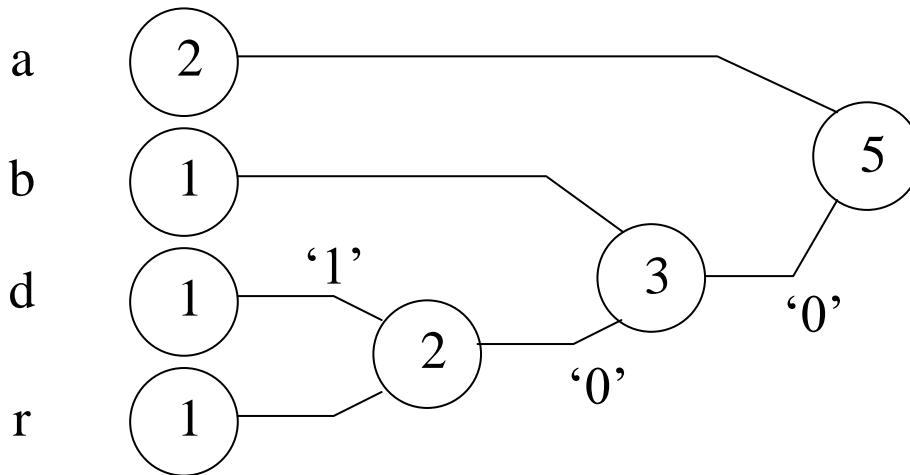


‘a’ codificado com ‘1’

$$I_5(a) = \log_2 (6/3) = 1 \text{ bit.}$$

Probabilidades e árvore alteradas

Exemplo: Huffman Adaptativo

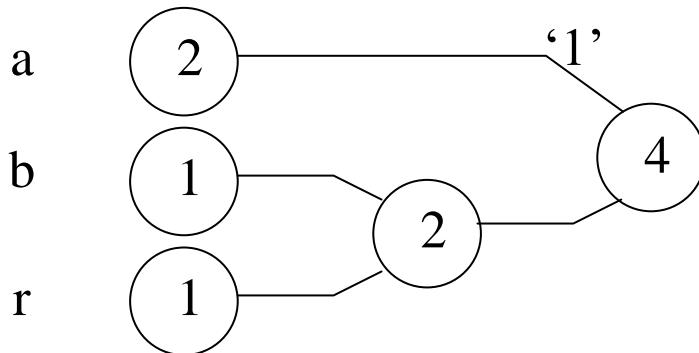


'd' codificado com '001'

$$I_6(d) = \log_2 (5/1) = 2,32 \text{ bits.}$$

Probabilidades e árvore alteradas

Exemplo: Huffman Adaptativo

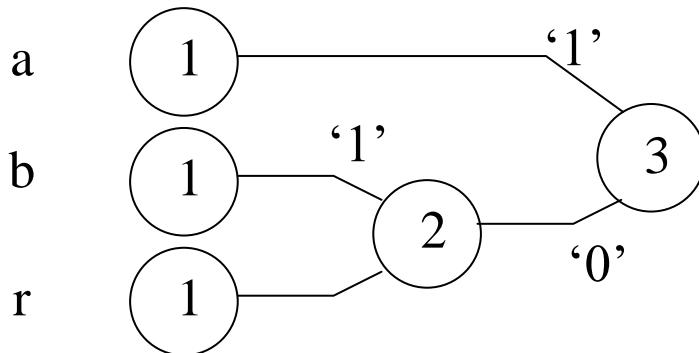


'a' codificado com '1'

$$I_7(a) = \log_2 (4/2) = 1 \text{ bit.}$$

Probabilidades e árvore alteradas

Exemplo: Huffman Adaptativo

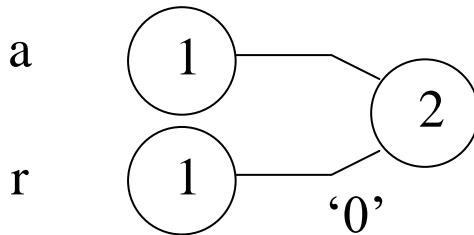


'b' codificado com '01'

$$I_8(b) = \log_2 (3/1) = 1,59 \text{ bit.}$$

Probabilidades e árvore alteradas

Exemplo: Huffman Adaptativo



'r' codificado com '0'

$$I_9(r) = \log_2 (2/1) = 1 \text{ bit.}$$

Probabilidades e árvore alteradas.

Exemplo: Huffman Adaptativo

'a' codificado com 0 bit

$$I_{10}(a) = \log_2 (1/1) = 0 \text{ bit}$$

Comprimento médio do código: $17/11=1,55$

Entropia?

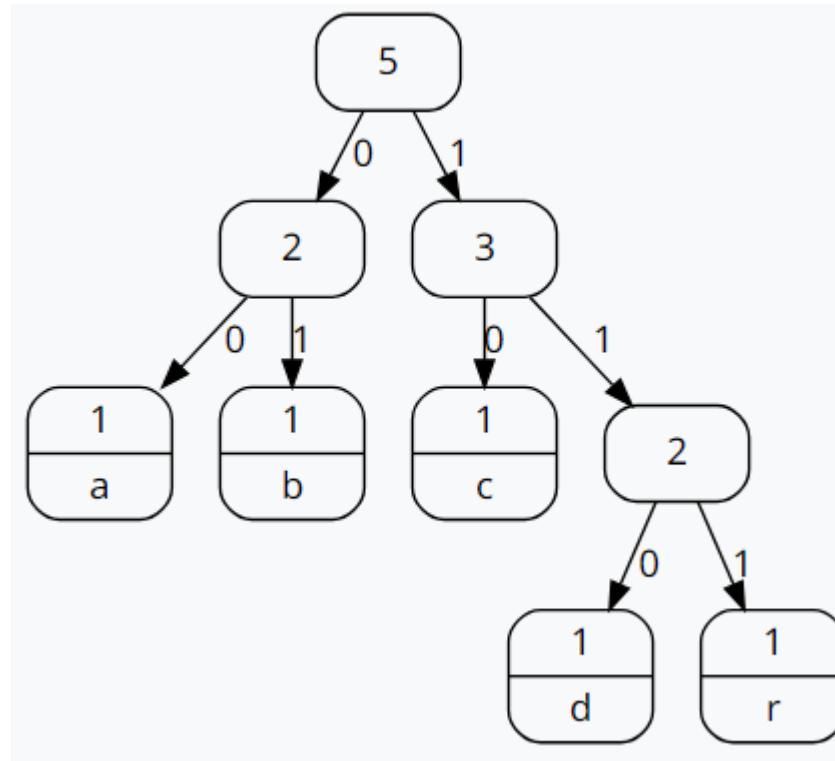
Entropia = 1,49

Exemplo: Huffman Adaptativo

Huffman adaptativo de incremento para abracadabra.

ρ : todos os símbolos não vistos, tidos como equiprováveis.

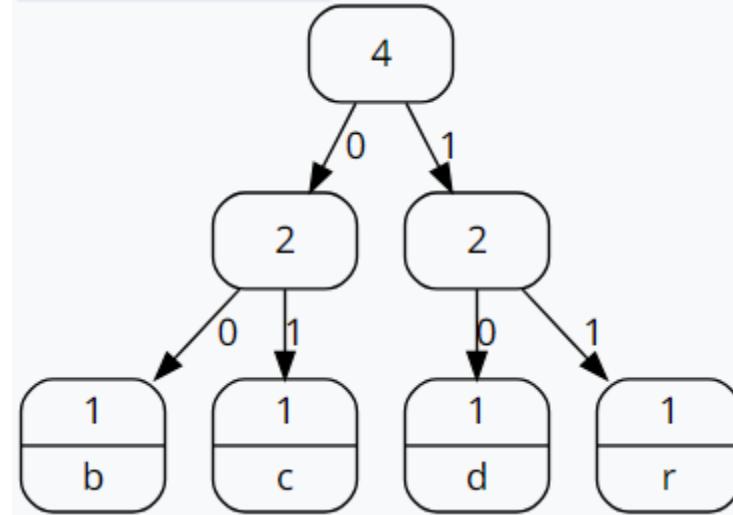
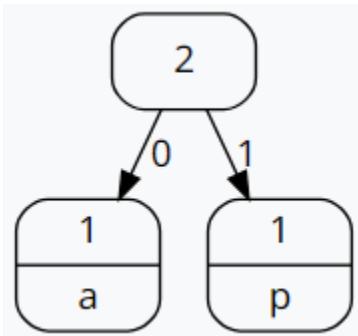
Exemplo: Huffman Adaptativo



abracadabra

Codifica 'p' com zero bit e 'a' com 00
00

Exemplo: Huffman Adaptativo

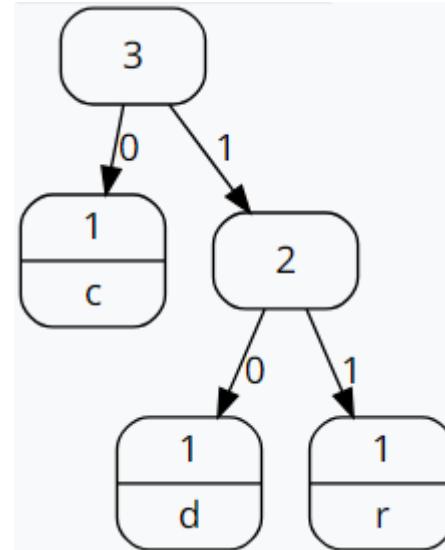
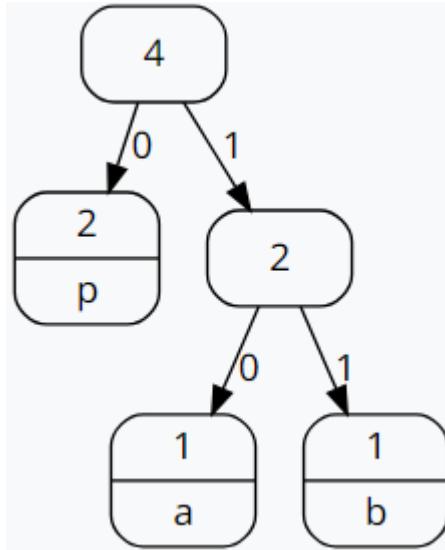


Abracadabra

Codifica 'p' com 1 e 'b' com 00

00100

Exemplo: Huffman Adaptativo



abracadabra

Codifica ‘p’ com 0 e ‘r’ com 11

00100011

Modelos Contextuais

Fonte S com $A = \{a_0, a_1, \dots, a_{M-1}\}$

Elemento x gerado por S é uma variável aleatória

Modelos não-contextuais: $P(x = a_i)$

Modelos contextuais: $P(x_n = a_i | x_{n-1}, x_{n-2}, \dots)$

Modelo de Markov de K -ésima ordem:

$$P(x_n = a_i | x_{n-1}, x_{n-2}, \dots, x_{n-K})$$

Modelos de memória finita

Modelos Contextuais

$$\mathbf{x}^K = x_{n-1}, x_{n-2}, \dots, x_{n-K}$$

Define-se informação condicional associada a a_i , dado um contexto \mathbf{x}^K como:

$$I(a_i | \mathbf{x}^K) = \log_2 \frac{1}{P(x_n = a_i | \mathbf{x}^K)}$$

Exemplo: modelo de ordem 1

Huffman semi-adaptativo contextual de ordem 1(um)

Alfabeto $A = \{a, b, c, d, r\}$

Mensagem: abracadabra

Exemplo: modelo de ordem 1

O símbolo ‘a’ ocorre 4 vezes como contexto de outro.

4 ocorrências do contexto a:

em 2 o símbolo subsequente é b

em 1 o símbolo subsequente é c

em 1 o símbolo subsequente é d

Portanto,

$$P(b | a) = 2/4 \Leftrightarrow I(b | a) = 1$$

$$P(c | a) = 1/4 \Leftrightarrow I(c | a) = 2$$

$$P(d | a) = 1/4 \Leftrightarrow I(d | a) = 2$$

Exemplo: modelo de ordem 1

Árvore de Huffman (. | a)

Analogamente:

$$P(r | b) = 1 \Leftrightarrow I(r | b) = 0$$

$$P(a | c) = 1 \Leftrightarrow I(a | c) = 0$$

$$P(a | d) = 1 \Leftrightarrow I(a | d) = 0$$

$$P(a | r) = 1 \Leftrightarrow I(a | r) = 0$$

Exemplo: modelo de ordem 1

Primeiro símbolo: codificar usando um modelo não contextual.

Comprimento médio atingido = $(1 + 1 + 1 + 2 + 2) / 11 = 0,6363$ bits/símbolo.

Modelos contextuais tendem a produzir muitos valores de informação condicional bem inferiores a 1(um), mas não nulos, e o Huffman torna-se ineficaz

Exemplo 2: modelo de ordem 1

Exemplo: Huffman semi-adaptativo
contextual de ordem 1

Mensagem:
abracadabraabraabarcabrababa

Modelos Contextuais

Exemplo: $\mathcal{A} = \{0,1\}$, Mensagem:
101010101010101010101010101010
1010101010101010101010101010...

Entropia ?

Modelo não-contextual

$H = 1 \text{ bit/símbolo}$ (1 bit/bit)

Modelo contextual de ordem 1

$H \approx 0 \text{ bit/símbolo}$

Modelos Contextuais

Se os símbolos dependem de um contexto de tamanho L , um modelo de Markov de ordem K , com $K \leq L$, deve ser superior (redução de entropia) a um modelo de ordem $K-1$.

Empregar modelos de ordem elevada para aumentar a compressão.

Problemas práticos limitam a ordem.

Modelos Contextuais

Número de contextos diferentes para modelo de ordem K e alfabeto com cardinalidade $|\mathcal{A}|$?

$|\mathcal{A}|^K$ contextos diferentes

Cresce exponencialmente com K

Quantos contadores passar ao decodificador (modelo semi-adaptativo)?

$|\mathcal{A}|^{K+1}$

Modelos Contextuais

Se $|\mathcal{A}| = 256$ e $K = 5$: aproximadamente um trilhão de contextos diversos.

Problemas práticos:

Memória, tempo de processamento,
estimativas de probabilidades,
armazenamento das probabilidades
condicionais.

Usar modelos adaptativos elimina apenas o último dos problemas citados!

Na prática, para $|\mathcal{A}| = 256$, $K = 0, 1, \dots, 10$

Huffman contextual adaptativo

Huffman adaptativo contextual de ordem 1

Alfabeto $\mathcal{A} = \{a, b, c, d, r\}$,

W: abracadabra

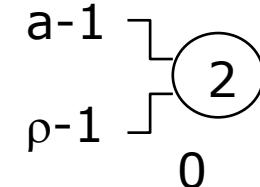
Hufmann contextual adaptativo

abra**cadabra**: ρ codificado com $P(\rho) = 1$;
‘a’ codificado em $K = -1$

$$W_c = 000$$

Modelo atualizado:

K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
				a	1	$\frac{1}{2}$	b – 00 c – 01 d – 10 r – 11
				ρ	1	$\frac{1}{2}$	



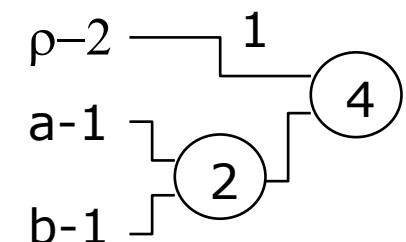
Hufmann contextual adaptativo

abracadabra: ρ codificado com $P(\rho) = \frac{1}{2}$;
b codificado em $K = -1$

$W_c = 000\ 0\ 00$

Modelo atualizado:

Contexto	K = 1			K = 0			K = -1 (Equip.)
	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	
a	b	1	$\frac{1}{2}$	a	1	$\frac{1}{4}$	c – 00 d – 01 r – 10
	ρ	1	$\frac{1}{2}$	b	1	$\frac{1}{4}$	
				ρ	2	$\frac{2}{4}$	



Hufmann contextual adaptativo

ab**r**acadabra: r codificado com $P(r) = 1/3$;
r codificado em $K = -1$ com $P(r) = 1/3$

$W_c = 000\ 0\ 00\ 1\ 10$

Modelo atualizado:

K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
a	b	1	$\frac{1}{2}$	a	1	$1/6$	c – 0 d – 1
	p	1	$\frac{1}{2}$	b	1	$1/6$	
b	r	1	$\frac{1}{2}$	r	1	$1/6$	
	p	1	$\frac{1}{2}$	p	3	$3/6$	

Hufmann contextual adaptativo

abracadabra: ‘a’ codificado com $P(a) = 1/6$

Modelo atualizado:

K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
a	b	1	$\frac{1}{2}$	a	2	$2/7$	c – 0 d – 1
	p	1	$\frac{1}{2}$	b	1	$1/7$	
b	r	1	$\frac{1}{2}$	r	1	$1/7$	c – 0 d – 1
	p	1	$\frac{1}{2}$	p	3	$3/7$	
r	a	1	$\frac{1}{2}$				
	p	1	$\frac{1}{2}$				

Hufmann contextual adaptativo

abra**c**adabra: $\rho|a$ codificado com $P(\rho|a) = \frac{1}{2}$

Em $K=0$, o símbolo a ser codificado não pode ser ‘b’, pois ele teria sido codificado no contexto ‘a’. Assim, ‘b’ é excluído da coluna $K = 0$, e as probabilidades nesta coluna são recalculadas, apenas para a codificação de ‘c’.

ρ codificado com $P(\rho) = 3/6$; ‘c’ codificado em $K= -1$ com $P(c)=1/2$

Hufmann contextual adaptativo

Modelo atualizado:

K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
a	b	1	$\frac{1}{4}$	a	2	$\frac{2}{9}$	d -
	c	1	$\frac{1}{4}$	b	1	$\frac{1}{9}$	
	p	2	$\frac{2}{4}$	c	1	$\frac{1}{9}$	
b	r	1	$\frac{1}{2}$	r	1	$\frac{1}{9}$	d -
	p	1	$\frac{1}{2}$	p	4	$\frac{4}{9}$	
r	a	1	$\frac{1}{2}$				
	p	1	$\frac{1}{2}$				

Hufmann contextual adaptativo

abracadabra: ‘a’ codificado com $P(a) = 2/9$

Modelo atualizado:

K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
a	b	1	$\frac{1}{4}$	a	3	$\frac{3}{10}$	d -
	c	1	$\frac{1}{4}$	b	1	$\frac{1}{10}$	
	p	2	$\frac{2}{4}$	c	1	$\frac{1}{10}$	
b	r	1	$\frac{1}{2}$	r	1	$\frac{1}{10}$	d -
	p	1	$\frac{1}{2}$	p	4	$\frac{4}{10}$	
c	a	1	$\frac{1}{2}$				
	p	1	$\frac{1}{2}$				
r	a	1	$\frac{1}{2}$				
	p	1	$\frac{1}{2}$				

Hufmann contextual adaptativo

abra**cadabra**: $\rho|a$ codificado com $P(\rho|a)=2/4$; ρ codificado com $P(\rho)=4/8$ (exclusão de ‘b’ e ‘c’) ; ‘d’ codificado em $K = -1$ com $P(d) =1$.

Não pode mais ocorrer ρ em $K = 0$, e $K=-1$ não será mais utilizado.

Hufmann contextual adaptativo

Modelo atualizado:

K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
a	b	1	1/6	a	3	3/7	-
	c	1	1/6	b	1	1/7	
	d	1	1/6	c	1	1/7	
	p	3	3/6	d	1	1/7	
b	r	1	½	r	1	1/7	-
	p	1	½				
c	a	1	½				-
	p	1	½				
r	a	1	½				-
	p	1	½				

Hufmann contextual adaptativo

abracadabra: ‘a’ codificado com $P(a) = 3/7$. Modelo atualizado:

K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
a	b	1	1/6	a	4	4/8	-
	c	1	1/6	b	1	1/8	
	d	1	1/6	c	1	1/8	
	p	3	3/6	d	1	1/8	
b	r	1	½	r	1	1/8	-
	p	1	½				
c	a	1	½				
	p	1	½				
d	a	1	½				-
	p	1	½				
r	a	1	½				
	p	1	½				

Hufmann contextual adaptativo

abracadabra: b|a codificado com
 $P(b|a)=1/6$. Modelo atualizado:

K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
a	b	2	2/7	a	4	4/9	-
	c	1	1/7	b	2	2/9	
	d	1	1/7	c	1	1/9	
	p	3	3/7	d	1	1/9	
b	r	1	½	r	1	1/9	-
	p	1	½				
c	a	1	½				
	p	1	½				
d	a	1	½				-
	p	1	½				
r	a	1	½				
	p	1	½				

Hufmann contextual adaptativo

abracadabra: r|b codificado com
 $P(r|b)=1/2$. Modelo atualizado:

K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
a	b	2	2/7	a	4	4/10	-
	c	1	1/7	b	2	2/10	
	d	1	1/7	c	1	1/10	
	p	3	3/7	d	1	1/10	
b	r	2	2/3	r	2	2/10	-
	p	1	1/3				
c	a	1	½				-
	p	1	½				
d	a	1	½				-
	p	1	½				
r	a	1	½				-
	p	1	½				

Hufmann contextual adaptativo

abracadabra: a|r codificado com
 $P(a|r)=1/2$. Modelo atualizado:

K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
a	b	2	2/7	a	5	5/11	-
	c	1	1/7	b	2	2/11	
	d	1	1/7	c	1	1/11	
	p	3	3/7	d	1	1/11	
b	r	2	2/3	r	2	2/11	-
	p	1	1/3				
c	a	1	½				-
	p	1	½				
d	a	1	½				-
	p	1	½				
r	a	2	2/3				-
	p	1	1/3				

Modelos de Markov de Ordem 1

Fonte com alfabeto $A = \{a_0, a_1, \dots, a_{M-1}\}$

Bigrama: par ordenado de símbolos consecutivos gerados pela fonte.

$$(a_i, a_j) \neq (a_j, a_i), \text{ para todo } i \neq j$$

Experimento probabilístico: observar bigrama.

Eventos de interesse:

E_i : a_i é o 1º elemento do bigrama

F_j : a_j é o 2º elemento do bigrama

Modelos de Markov de Ordem 1

Probabilidade de ocorrência simultânea dos eventos E_i e F_j :

$$P(E_i, F_j) = P(F_j, E_i)$$

Probabilidade de ocorrência de F_j dada a ocorrência de E_i :

$$P(F_j | E_i) \neq P(E_i | F_j)$$

Modelos de Markov de Ordem 1

Regra da probabilidade condicional

$$P(F_j \mid E_i) = \frac{P(E_i, F_j)}{P(E_i)}$$

Modelos de Markov de Ordem 1

Exemplo

$W = \text{abracadabraabraabarcabrabab}$

$\mathcal{A} = \{a, b, c, d, r\}$

Eventos:

F_c : símbolo ‘c’ na 2^a posição de um bigrama

E_a : símbolo ‘a’ na 1^a posição de um bigrama

Calcular $P(E_a, F_c)$, $P(F_c, E_a)$, $P(E_a/F_c)$, $P(F_c/E_a)$

Modelos de Markov de Ordem 1

Estimar probabilidades dos eventos pela frequência relativa

$$P(E_a) = ? \quad P(F_c) = ?$$

W: 26 símbolos, 25 bigramas (ab, br, ra, ...ba)

E_a ocorre 11 vezes na mensagem

F_c ocorre 2 vezes

$$P(E_a) = 11/25$$

$$P(F_c) = 2/25$$

Modelos de Markov de Ordem 1

$$P(F_c | E_a) = ? \quad P(E_a | F_c) = ? \dots$$

$$P(E_a, F_c) = P(F_c, E_a) = ?$$

E_a ocorreu 11 vezes; destas, em uma F_c ocorreu: $P(F_c | E_a) = 1/11$

F_c ocorreu 2 vezes; destas, em uma E_a ocorreu: $P(E_a | F_c) = 1/2$

E_a e F_c ocorreram simultaneamente 1 vez nos 25 bigramas: $P(E_a, F_c) = P(E_a, F_c) = 1/25$

Modelos de Markov de Ordem 1

Observe que:

$$P(F_c | E_a) = \frac{P(E_a, F_c)}{P(E_a)} = \frac{1/25}{11/25} = \frac{1}{11}$$

$$P(E_a | F_c) = \frac{P(E_a, F_c)}{P(F_c)} = \frac{1/25}{2/25} = \frac{1}{2}$$

Modelos de Markov de Ordem 1

Distribuição de probabilidades conjuntas:

1º símbolo do bigrama	2º símbolo do bigrama				
	a	b	c	d	r
a	2/25	6/25	1/25	1/25	1/25
b	2/25	0	0	0	4/25
c	2/25	0	0	0	0
d	1/25	0	0	0	0
r	4/25	0	1/25	0	0

O somatório de todas as células da tabela resulta em 1

Modelos de Markov de Ordem 1

Distribuição de probabilidades condicionais $P(F_j | E_i)$

1º símbolo do bigrama	2º símbolo do bigrama				
	a	b	c	d	r
a	2/11	6/11	1/11	1/11	1/11
b	2/6	0	0	0	4/6
c	2/2	0	0	0	0
d	1/1	0	0	0	0
r	4/5	0	1/5	0	0

O somatório de todas as células de cada linha da tabela resulta em 1

Entropia de Ordem 2

Markov de ordem 1: entropia de ordem 2

$I(F_j|E_i) = \log_2[1/P(F_j|E_i)]$: informação associada a a_j dado o contexto a_i

W = abracadabraabraababarcabraba,

$\mathcal{A} = \{a, b, c, d, r\}$

$$H_2 = [I(b|a) + I(r|b) + I(a|r) \dots + I(a|b)]/25$$

$$H_2 = [2I(a|a) + 6I(b|a) + \dots + 1I(c|r)]/25$$

Entropia de Ordem 2

N_{ij} : nº de ocorrências de a_j no contexto a_i
= nº de ocorrências de $a_i a_j$

N_b : total de bigramas em uma mensagem

$$\begin{aligned} H_2 = & -\frac{1}{N_b} \times [N_{00} \log_2 P(F_0 | E_0) + \\ & + N_{01} \log_2 P(F_1 | E_0) + N_{02} \log_2 P(F_2 | E_0) + \\ & \dots + N_{M-1,M-1} \log_2 P(F_{M-1} | E_{M-1})] \end{aligned}$$

Entropia de Ordem 2

Estimando probabilidades por freqüência relativa:

$$\begin{aligned} H_2 = & -[P(E_0, F_0) \log_2 P(F_0 | E_0) + \\ & + P(E_0, F_1) \log_2 P(F_1 | E_0) + \dots \\ & + P(E_{M-1}, F_{M-1}) \log_2 P(F_{M-1} | E_{M-1})] \end{aligned}$$

$$H_2 = -\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} P(E_i, F_j) \log_2 P(F_j | E_i)$$

Entropia de Ordem N

Contexto v_i : $N-1$ símbolos consecutivos

Alfabeto $\mathcal{A} = \{a_0, a_1, \dots, a_{M-1}\}$: $C = M^{N-1}$
contextos distintos

$$v_0 = a_0 \ a_0 \ a_0 \dots a_0$$

$$v_1 = a_0 \ a_0 \ a_0 \dots a_1$$

...

$$v_{C-1} = a_{M-1} \ a_{M-1} \ a_{M-1} \dots a_{M-1}$$

Entropia de Ordem N

Eventos E_i e F_j

E_i : o contexto v_i ocorre no início de um N-grama

F_j : o símbolo a_j ocorre na última posição de um N-grama

Mesmas equações de probabilidades condicionais já apresentadas.

Entropia de Ordem N

Distribuição de probabilidades conjuntas:

Contexto	símbolo			
	a_0	a_1	...	a_{M-1}
v_0				
v_1				
...				
v_{C-1}				

O somatório de todas as células da tabela resulta em 1

Entropia de Ordem N

Distribuição de probabilidades condicionais $P(F_j | E_i)$

Contexto	símbolo			
	a_0	a_1	...	a_{M-1}
v_0				
v_1				
...				
v_{C-1}				

O somatório de todas as células de cada linha da tabela resulta em 1

Entropia de Ordem N

Entropia de ordem N

$$H_N = - \sum_{i=0}^{C-1} \sum_{j=0}^{M-1} P(E_i, F_j) \log_2 P(F_j | E_i)$$

Entropia (fonte estacionária)

$$H = \lim_{N \rightarrow \infty} H_N$$

PPM

Prediction by Partial Matching
Cleary, J.G. and Witten, I.H. (1984)
“Data compression using adaptive
coding and partial string matching,”
IEEE Transactions on
Communications, 32(4), 396-402

Exemplo: PPM

$W = \text{abracadabra}$

Contexto máximo $K = 2$

$\mathcal{A} = \{a, b, c, d, r\}$

Exemplo: PPM

Modelo inicial (vazio) :

Exemplo: PPM

abracadabra: ‘a’ codificado em $K = -1$
com $P(a) = 1/5$

Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
								a	1	1/2	1/4
								p	1	1/2	

Exemplo: PPM

abracadabra: ρ codificado com $P(\rho) = \frac{1}{2}$;
'b' em $K = -1$ com $P(b) = \frac{1}{4}$

Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
				a	b	1	1/2	a	1	1/4	1/3
					ρ	1	1/2	b	1	1/4	
								ρ	2	2/4	

Exemplo: PPM

ab**r**acadabra: ‘**r**’ codificado com $P(r) = \frac{1}{2}$; ‘r’ em K = -1 com $P(r) = 1/3$.

Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	1	1/2	a	b	1	1/2	a	1	1/6	1/2
	ρ	1	1/2		ρ	1	1/2	b	1	1/6	
				b	r	1	1/2	r	1	1/6	
					ρ	1	1/2	ρ	3	3/6	

Exemplo: PPM

abra**cadabra**: ‘a’ codificado com
 $P(a)=1/6$

Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	1	1/2	a	b	1	1/2	a	2	2/7	1/2
	ρ	1	1/2		ρ	1	1/2	b	1	1/7	
br	a	1	1/2	b	r	1	1/2	r	1	1/7	1/2
	ρ	1	1/2		ρ	1	1/2	ρ	3	3/7	
				r	a	1	1/2				
				ρ	1	1/2					

Exemplo: PPM

abracadabra: $\rho|a$ codificado com
 $P(\rho|a)=1/2$.

Em $K = 0$, o símbolo a ser codificado não pode ser ‘b’, pois ele teria sido codificado no contexto ‘a’. Assim, ‘b’ é excluído da coluna $K = 0$, e as probabilidades nesta coluna são recalculadas, apenas para a codificação de ‘c’.

ρ codificado com prob. $3/6$, e ‘c’ codificado em $K = -1$ com $P(c)=1/2$.

Exemplo: PPM

Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	1	1/2	a	b	1	1/4	a	2	2/9	1
	p	1	1/2		c	1	1/4	b	1	1/9	
br	a	1	1/2		p	2	2/4		1	1/9	1
	p	1	1/2	b	r	1	1/2	c	1	1/9	
ra	c	1	1/2		p	1	1/2	r	1	1/9	1
	p	1	1/2	r	a	1	1/2	p	4	4/9	
					p	1	1/2				

Exemplo: PPM

abracadabra: ‘a’ codificado com
 $P(a)=2/9$.

Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	1	1/2	a	b	1	1/4	a	3	3/10	1
	p	1	1/2		c	1	1/4	b	1	1/10	
ac	a	1	1/2	b	p	2	2/4		1	1/10	1
	p	1	1/2		r	1	1/2	c	1	1/10	
br	a	1	1/2	c	p	1	1/2	r	1	1/10	1
	p	1	1/2		a	1	1/2	p	4	4/10	
ra	c	1	1/2	r	p	1	1/2				1
	p	1	1/2		a	1	1/2				
					p	1	1/2				

Exemplo: PPM

abracadabra: $\rho|a$ codificado com $P(\rho|a)=2/4$; ρ codificado com $P(\rho)=4/8$ (exclusão de 'b' e 'c') ; 'd' em $K = -1$ com $P(d)=1$.

Não pode mais ocorrer ρ em $K=0$, e $K=-1$ não será mais utilizado.

Exemplo: PPM

Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	1	1/2	a	b	1	1/6	a	3	3/7	-
	p	1	1/2		c	1	1/6	b	1	1/7	
ac	a	1	1/2		d	1	1/6		1	1/7	
	p	1	1/2		p	3	3/6	c	1	1/7	
br	a	1	1/1	b	r	1	1/2	d	1	1/7	-
	p	1	1/2		p	1	1/2	r	1	1/7	
ca	d	1	1/2	c	a	1	1/2				
	p	1	1/2		p	1	1/2				
ra	c	1	1/2	r	a	1	1/2				-
	p	1	1/2		p	1	1/2				

Exemplo: PPM

abracadabra: 'a' codificado com
 $P(a)=3/7$. Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	1	1/2	a	b	1	1/6	a	4	4/8	-
	p	1	1/2		c	1	1/6	b	1	1/8	
ac	a	1	1/2		d	1	1/6		1	1/8	
	p	1	1/2		p	3	3/6	c	1	1/8	
ad	a	1	1/2	b	r	1	1/2	d	1	1/8	-
	p	1	1/2		p	1	1/2	r	1	1/8	
br	a	1	1/2	c	a	1	1/2	d	1	1/8	
	p	1	1/2		p	1	1/2		1	1/8	
ca	d	1	1/2	d	a	1	1/2	r	1	1/8	
	p	1	1/2		p	1	1/2		1	1/8	
ra	c	1	1/2	r	a	1	1/2	-	-	-	-
	p	1	1/2		p	1	1/2		-	-	

Exemplo: PPM

abracadabra: b|a codificado com
 $P(b|a)=1/6$.

Exemplo: PPM

Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	1	1/2	a	b	2	2/7	a	4	4/9	-
	p	1	1/2		c	1	1/7	b	2	2/9	
ac	a	1	1/2		d	1	1/7		1	1/9	
	p	1	1/2		p	3	3/7	c	1	1/9	
ad	a	1	1/2	b	r	1	1/2	d	1	1/9	
	p	1	1/2		p	1	1/2	r	1	1/9	
br	a	1	1/2	c	a	1	1/2				-
	p	1	1/2		p	1	1/2				
ca	d	1	1/2	d	a	1	1/2				
	p	1	1/2		p	1	1/2				
da	b	1	1/2	r	a	1	1/2				
	p	1	1/2		p	1	1/2				
ra	c	1	1/2								-
	p	1	1/2								

Exemplo: PPM

abracadabra: r|ab codificado com
 $P(r|ab)=1/2$.

Modelo atualizado:

Exemplo: PPM

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	2	2/3	a	b	2	2/7	a	4	4/10	-
	p	1	1/3		c	1	1/7	b	2	2/10	
ac	a	1	1/2		d	1	1/7		1	1/10	
	p	1	1/2	b	p	3	3/7	d	1	1/10	
ad	a	1	1/2		r	2	2/3		2	2/10	
	p	1	1/2	c	p	1	1/3	r	2	1/10	
br	a	1	1/2		a	1	1/2		1	1/10	
	p	1	1/2	c	p	1	1/2		2	2/10	
ca	d	1	1/2	d	a	1	1/2	d	1	1/10	
	p	1	1/2		p	1	1/2		2	2/10	
da	b	1	1/2	r	a	1	1/2	r	1	1/10	
	p	1	1/2		p	1	1/2		2	2/10	
ra	c	1	1/2								-
	p	1	1/2								

Exemplo: PPM

abracadabra: a|br codificado com
 $p(a|br)=1/2$.

Modelo atualizado:

Exemplo: PPM

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	2	2/3	a	b	2	2/7	a	5	5/11	-
	p	1	1/3		c	1	1/7	b	2	2/11	
ac	a	1	1/2		d	1	1/7		1	1/11	
	p	1	1/2		p	3	3/7	c	1	1/11	
ad	a	1	1/2	b	r	2	2/3	d	1	1/11	
	p	1	1/2		p	1	1/3	r	2	2/11	
br	a	2	2/3	c	a	1	1/2				-
	p	1	1/3		p	1	1/2				
ca	d	1	1/2	d	a	1	1/2				
	p	1	1/2		p	1	1/2				
da	b	1	1/2	r	a	2	2/3				
	p	1	1/2		p	1	1/3				
ra	c	1	1/2								-
	p	1	1/2								

Codificação Aritmética

Mensagem: assassinar, alfabeto = {a, s, i, n, r}

Probabilidades (modelo semi-adaptativo não contextual)

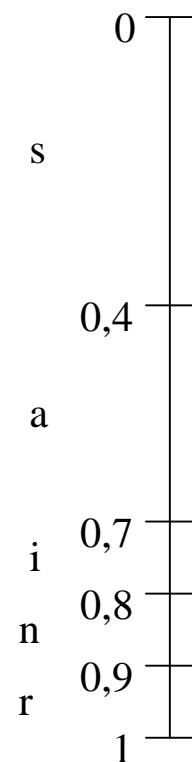
$$s - 0,4$$

$$a - 0,3$$

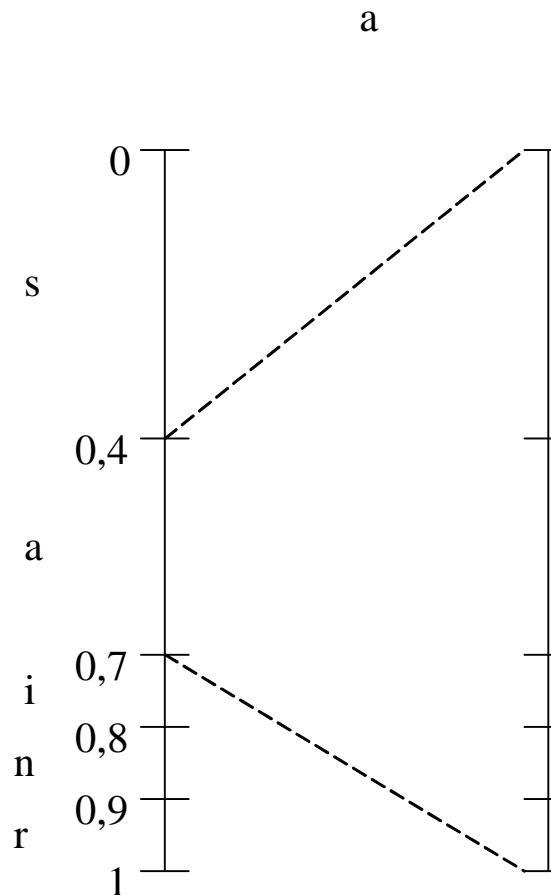
$$i - 0,1$$

$$n - 0,1$$

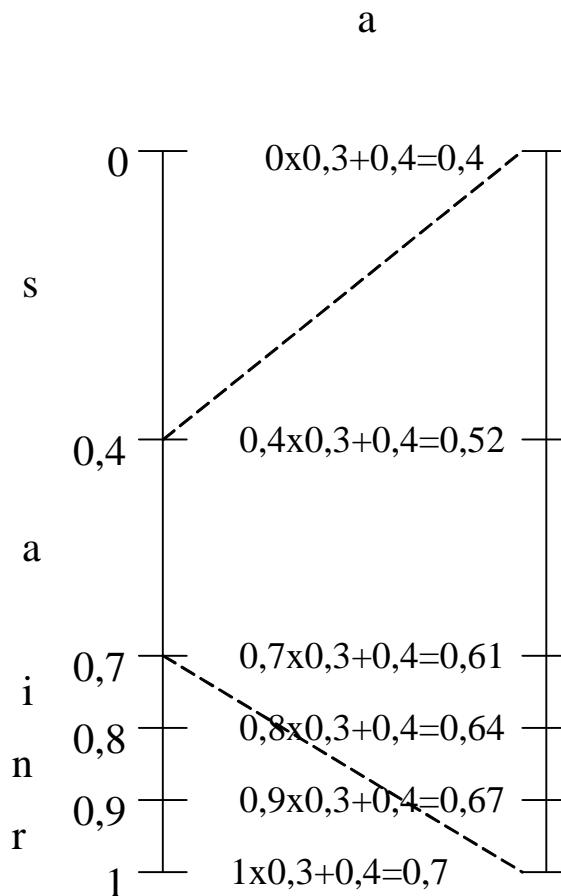
$$r - 0,1$$



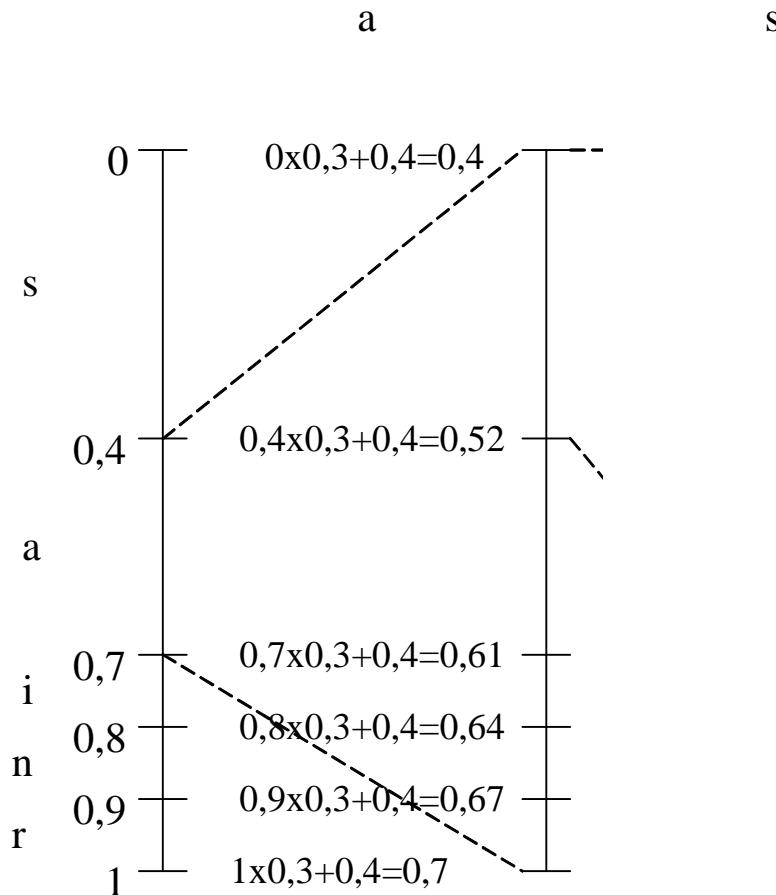
Codificação Aritmética



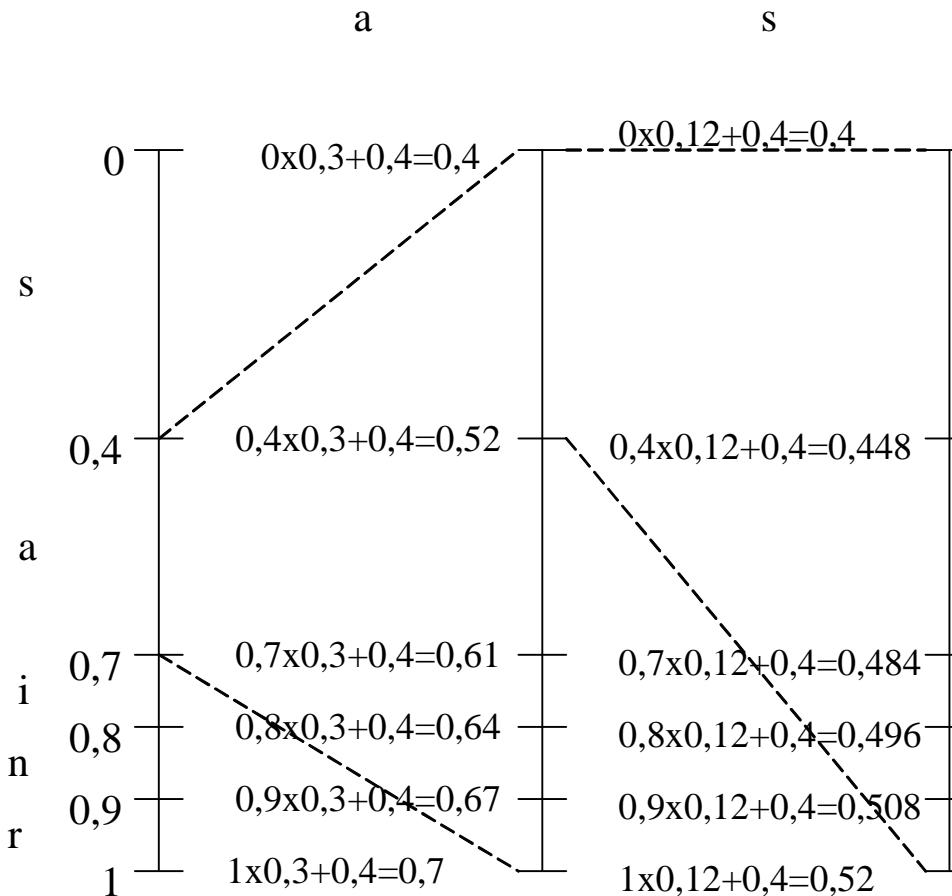
Codificação Aritmética



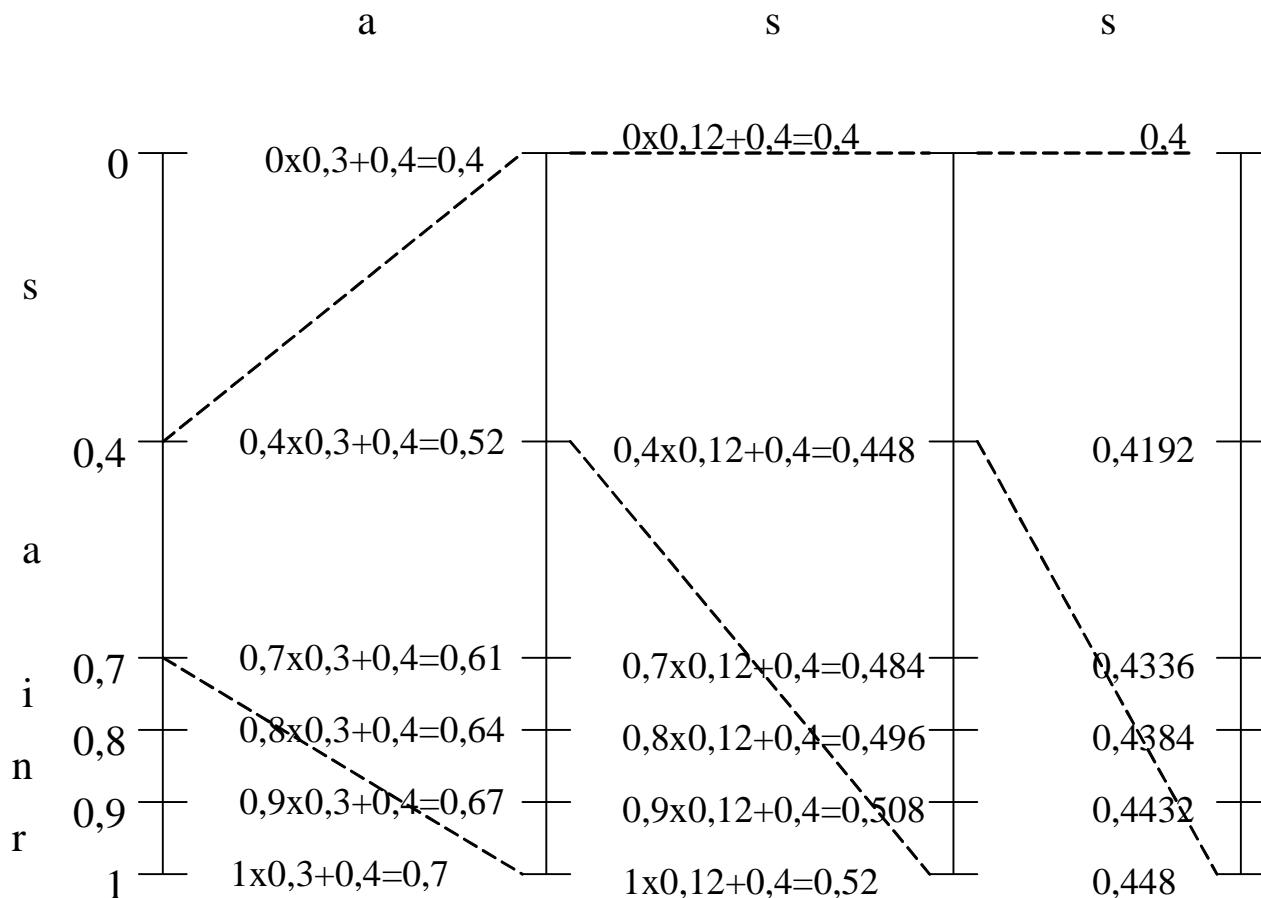
Codificação Aritmética



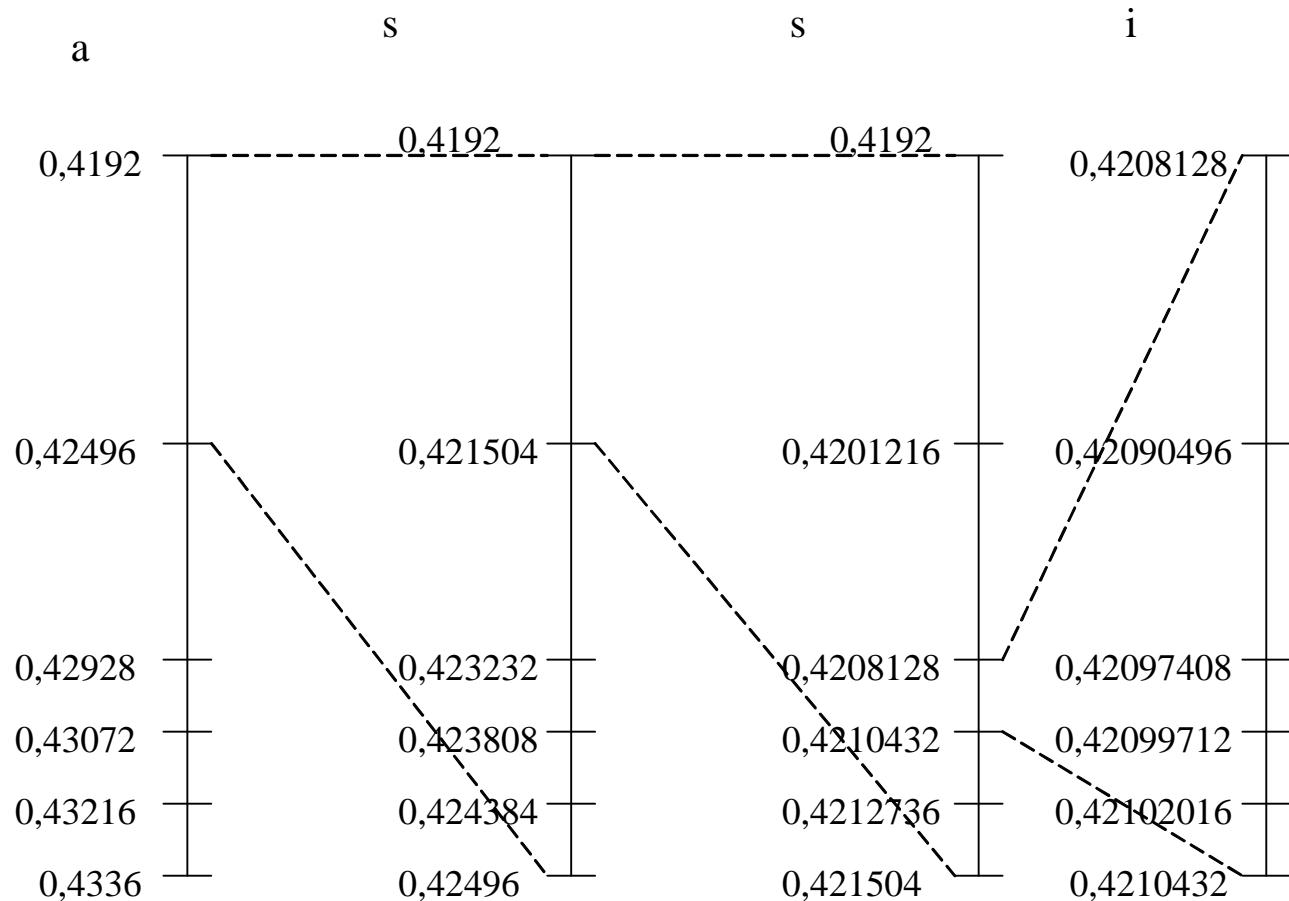
Codificação Aritmética



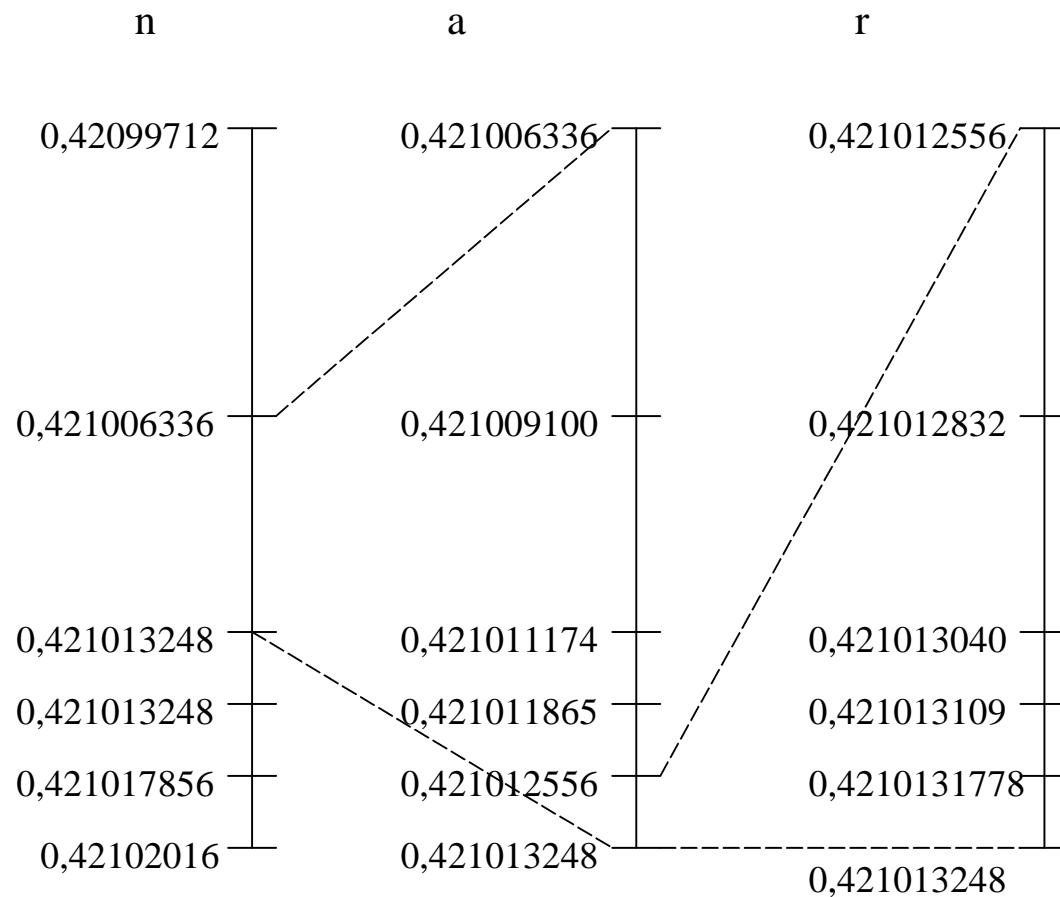
Codificação Aritmética



Codificação Aritmética



Codificação Aritmética



Mensagem Codificada: casas decimais de um número no intervalo final com o menor número possível de casas decimais: 421013

Codificação Aritmética

Quando os limites inferior e superior do intervalo atual coincidem na 1^a casa decimal, esta não muda mais, e o dígito correspondente já pode ser enviado para a saída.

Após a coincidência na 1^a casa decimal, a mesma observação vale para a 2^a casa, e assim sucessivamente.

Após coincidência na 1^a casa decimal dos limites do intervalo atual, pode-se multiplicá-los por 10 e jogar para a saída a parte inteira.

Decodificação Aritmética

O número n que representa a mensagem codificada está no intervalo associado ao 1º símbolo codificado. No exemplo, 0,421013 está no intervalo do “a”. Decodifica-se assim o 1º símbolo.

Decodificação do segundo símbolo:
retirar de n o efeito do primeiro símbolo.

Decodificação Aritmética

Como o primeiro símbolo, x , altera o intervalo inicial I ?

Multiplica os limites de I por $P(x)$ e adiciona aos limites de I o limite inferior do intervalo atribuído a x

Reverter o efeito: subtrair do número que representa a mensagem codificada o limite inferior do intervalo atribuído a x , e dividir o resultado por $P(x)$

Decodificação Aritmética

Decodificação de ‘s’, no exemplo:

$$\text{Número} = (0,421013 - 0,4)/0,3 = \\ 0,070043333$$

O efeito do segundo (terceiro etc) símbolo pode ser retirado do número resultante da mesma forma.

Decodificação Aritmética

Número	Símbolo	ls_i	rs_i
0.421013	a	0.4	0.3
0.070043333	s	0.0	0.4
0.175108333	s	0.0	0.4
0.437770833	a	0.4	0.3
0.125902777	s	0.0	0.4
0.314756944	s	0.0	0.4
0.786892361	i	0.7	0.1
0.86892361	n	0.8	0.1
0.689236108	a	0.4	0.3
0.96412036	r	0.9	0.1

Codificação Aritmética e entropia

Codificação aritmética atinge a entropia?

Independência:

$$P(A, B) = P(A|B) \cdot P(B) = P(A) \cdot P(B)$$

O intervalo final tem tamanho igual ao produto das probabilidades de cada símbolo na mensagem W , que é a probabilidade de ocorrência da mensagem. $P(W)$ considerando-se símbolos independentes.

Codificação Aritmética e entropia

No exemplo:

$$P(W) =$$

$$0,3 \times 0,4 \times 0,4 \times 0,3 \times 0,4 \times 0,4 \times 0,1 \times 0,1 \times 0,3 \times 0,1 = \\ 0,000000692 = 0,421013248 - \\ 0,421012556$$

O código terá um número de dígitos igual ao número de zeros, de $P(W)$, em sequência após a vírgula decimal, e no máximo mais um dígito.

Codificação Aritmética e entropia

Exemplo de uma situação em que seria necessário um dígito adicional:

$$0,421013248 - 0,421013148 = \\ 0,0000001.$$

Neste caso, a mensagem codificada seria 4210132.

Codificação Aritmética e entropia

Informação (em dígitos decimais) associada à mensagem W:

$$I_{dec} = -\log_{10}P(W)$$

Escrevendo-se $P[W]$ na forma $0,abc\dots 10^{-m}$, com ‘a’ não nulo:

$$\begin{aligned} I_{dec} &= -\log_{10}(0,abc\dots 10^{-m}) = \\ &= -\log_{10}(0,abc\dots) - \log_{10}10^{-m} = \\ &= m - \log_{10}(0,abc\dots) \end{aligned}$$

Em que $-\log_{10}(0,abc\dots) \in (0 \text{ e } 1]$

Codificação Aritmética e entropia

O total de bits de W na codificação ótima da mensagem é igual ao número de zeros em sequência após a vírgula decimal da probabilidade da mensagem (m), e no máximo mais um dígito.

A codificação aritmética atribui no máximo $-\log_{10}P[W] + 1$ bit à mensagem inteira!

O mesmo ocorre para modelos contextuais, substituindo-se probabilidades a priori por condicionais.

Codificação Aritmética e entropia

No exemplo:

$$I_{dec} = -\log_{10} P(W) = -\log_{10}[0,6912 \times 10^{-6}] = \\ 6,16 \text{ dígitos decimais.}$$

Relação entre I_{dec} e I_{bin}

$$I_{bin} = -\log_2 P(W) = -\log_2[0,6912 \times 10^{-6}] = \\ 20,47 \text{ bits}$$

$$I_{bin}/I_{dec} = 3.32 = \log_2 10$$

Codificação Baseada em Dicionário

Ziv e Lempel, 1976: Complexidade de sistemas dinâmicos

Ziv e Lempel, 1977: LZ77

Ziv e Lempel, 1978: LZ78

Welch, 1984: LZW

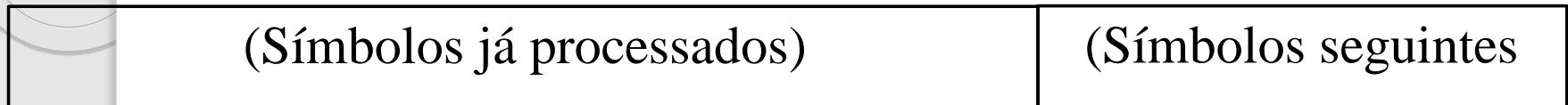
LZA, LZM, LZFA, ...

LZ77

D-1

0

L-1



o que eles foram e me fizeram. Olhos de de ressaca. É o

.....14..... 2,1,0

“de ressaca” codificado como (distância, comprimento) = (14,10)

foram e me fizeram. Olhos de ressaca? Vá, de É o que me dá i

Exemplo LZW

$W = \text{abracadabra}$; alfabeto = {a, b, c, d, r}

Codificação

Dicionário inicial:

0 a

1 b

2 c

3 d

4 r

Exemplo LZW (Cont.)

W = **a**bracadabra

Frase 'a' está no dicionário

Frase 'ab' não está no dicionário

Mensagem codificada = 0 (última frase encontrada no dicionário)

Dicionário:

0 a 5 ab

1 b

2 c

3 d

4 r

Exemplo LZW (Cont.)

W = abracadabra

‘b’ está no dicionário; ‘br’ não está no dicionário

Mensagem codificada = 0 1

Dicionário:

0 a 5 ab

1 b 6 br

2 c

3 d

4 r

Exemplo LZW (Cont.)

W = ab**r**acadabra

‘r’ está no dicionário; ‘ra’ não está no dicionário

Mensagem codificada = 0 1 4

Dicionário:

0 a 5 ab

1 b 6 br

2 c 7 ra

3 d

4 r

Exemplo LZW (Cont.)

W = abra**a**cadabra

‘a’ está no dicionário; ‘ac’ não está no dicionário

Mensagem codificada = 0 1 4 0

Dicionário:

0 a 5 ab

1 b 6 br

2 c 7 ra

3 d 8 ac

4 r

Exemplo LZW (Cont.)

W = abracadabra

‘c’ está no dicionário; ‘ca’ não está no dicionário

Mensagem codificada = 0 1 4 0 2

Dicionário:

0 a 5 ab

1 b 6 br

2 c 7 ra

3 d 8 ac

4 r 9 ca

Exemplo LZW (Cont.)

W = abracadabra

‘a’ está no dicionário; ‘ad’ não está no dicionário

Mensagem codificada = 0 1 4 0 2 0

Dicionário:

0 a 5 ab 10 ad

1 b 6 br

2 c 7 ra

3 d 8 ac

4 r 9 ca

Exemplo LZW (Cont.)

W = abracadabra

‘d’ está no dicionário; ‘da’ não está no dicionário

Mensagem codificada = 0 1 4 0 2 0 3

Dicionário:

0 a 5 ab 10 ad

1 b 6 br 11 da

2 c 7 ra

3 d 8 ac

4 r 9 ca

Exemplo LZW (Cont.)

W = abracadabra

‘a’ está no dicionário; ‘ab’ está no dicionário; ‘abr’ não está no dicionário

Mensagem codificada = 0 1 4 0 2 0 3 5

Dicionário:

0	a	5	ab	10	ad
1	b	6	br	11	da
2	c	7	ra	12	abr
3	d	8	ac		
4	r	9	ca		

Exemplo LZW (Cont.)

W = abracadab**ra**

‘r’ está no dicionário; ‘ra’ está no dicionário; EOF

Mensagem codificada = 0 1 4 0 2 0 3 5 7

Dicionário:

0 a 5 ab 10 ad

1 b 6 br 11 da

2 c 7 ra 12 abr

3 d 8 ac 13 raEOF

4 r 9 ca

Exemplo LZW (Cont.)

Decodificação

W_c : 0 1 4 0 2 0 3 5 7

Dicionário inicial:

0 a

1 b

2 c

3 d

4 r

Exemplo LZW (Cont.)

W_c : 0 1 4 0 2 0 3 5 7

W : a

Dicionário :

0 a 5 a?

1 b

2 c

3 d

4 r

Exemplo LZW (Cont.)

W_c : 0 1 4 0 2 0 3 5 7

W : ab

Dicionário :

0 a 5 ab

1 b 6 b?

2 c

3 d

4 r

Exemplo LZW (Cont.)

W_c : 0 1 **4** 0 2 0 3 5 7

W : abr

Dicionário :

0 a 5 ab

1 b 6 br

2 c 7 r?

3 d

4 r

Exemplo LZW (Cont.)

W_c : 0 1 4 **0** 2 0 3 5 7

W : abra

Dicionário :

0 a 5 ab

1 b 6 br

2 c 7 ra

3 d 8 a?

4 r

Exemplo LZW (Cont.)

W_c : 0 1 4 0 **2** 0 3 5 7

W : abrac

Dicionário :

0 a 5 ab

1 b 6 br

2 c 7 ra

3 d 8 ac

4 r 9 c?

Exemplo LZW (Cont.)

W_c : 0 1 4 0 2 0 3 5 7

W : abraca

Dicionário :

0 a 5 ab 10 a?

1 b 6 br

2 c 7 ra

3 d 8 ac

4 r 9 ca

Exemplo LZW (Cont.)

W_c : 0 1 4 0 2 0 **3** 5 7

W: abracad

Dicionário :

0 a 5 ab 10 ad

1 b 6 br 11 d?

2 c 7 ra

3 d 8 ac

4 r 9 ca

Exemplo LZW (Cont.)

W_c : 0 1 4 0 2 0 3 **5** 7

W: abracadab

Dicionário :

0 a 5 ab 10 ad

1 b 6 br 11 da

2 c 7 ra 12 ab?

3 d 8 ac

4 r 9 ca

Exemplo LZW (Cont.)

W_c : 0 1 4 0 2 0 3 5 7

W: abracadabra

Dicionário :

0 a 5 ab 10 ad

1 b 6 br 11 da

2 c 7 ra 12 abr

3 d 8 ac 13 raEOF

4 r 9 ca

LZW - Exemplo 2

$W = aabababaaa$; alfabeto = {a, b}

Codificação

Dicionário inicial:

0 a

1 b

LZW - Exemplo 2 (Cont.)

$W = \text{aabababaaa}$

‘a’ está no dicionário; ‘aa’ não está no dicionário

Mensagem codificada: 0

Dicionário:

0 a

1 b

2 aa

LZW - Exemplo 2 (Cont.)

$W = a\textcolor{red}{a}bababaaa$

‘a’ está no dicionário; ‘ab’ não está no dicionário

Mensagem codificada: 0 0

Dicionário:

0 a

1 b

2 aa

3 ab

LZW - Exemplo 2 (Cont.)

$W = aabababaaa$

‘b’ está no dicionário; ‘ba’ não está no dicionário

Mensagem codificada: 0 0 1

Dicionário:

0 a 4 ba

1 b

2 aa

3 ab

LZW - Exemplo 2 (Cont.)

W = aab**ab**abaaa

‘a’ está no dicionário; ‘ab’ está no dicionário; ‘aba’ não está no dicionário

Mensagem codificada: 0 0 1 3

Dicionário:

0 a 4 ba

1 b 5 aba

2 aa

3 ab

LZW - Exemplo 2 (Cont.)

W = aabab**aba**aa

'a' está no dic.; 'ab' está no dic.; 'aba' está no dic.; 'abaa' não está no dic.

Mensagem codificada: 0 0 1 3 5

Dicionário:

0 a 4 ba

1 b 5 aba

2 aa 6 abaa

3 ab

LZW - Exemplo 2 (Cont.)

W = aabababaaa

'a' está no dic.; 'aa' está no dic.; aaEOF
não está no dic.

Mensagem codificada: 0 0 1 3 5 2

Dicionário:

0 a 4 ba

1 b 5 aba

2 aa 6 abaa

3 ab 7 aaEOF

LZW - Exemplo 2 (Cont.)

Decodificação

W_c : 0 0 1 3 5 2

Dicionário inicial:

0 a

1 b

LZW - Exemplo 2 (Cont.)

W_c : 0 0 1 3 5 2

W : a

Dicionário:

0 a

1 b

2 a?

LZW - Exemplo 2 (Cont.)

W_c : 0 **0** 1 3 5 2

W : aa

Dicionário:

0 a

1 b

2 aa

3 a?

LZW - Exemplo 2 (Cont.)

W_c : 0 0 1 3 5 2

W : aab

Dicionário:

0 a 4 b?

1 b

2 aa

3 ab

LZW - Exemplo 2 (Cont.)

W_c : 0 0 1 **3** 5 2

W : aabab

Dicionário:

0 a 4 ba

1 b 5 ab?

2 aa

3 ab

LZW - Exemplo 2 (Cont.)

W_c : 0 0 1 3 **5** 2

Frase 5 incompleta!

O 1º símbolo da frase em 5 é o último símbolo da última frase inserida (5):

W : aabababa

Dicionário:

0 a 4 ba

1 b 5 aba

2 aa 6 aba?

3 ab

LZW - Exemplo 2 (Cont.)

W_c : 0 0 1 3 5 **2**

W : aabababaaa

Dicionário:

0 a 4 ba

1 b 5 aba

2 aa 6 abaa

3 ab 7 aaEOF

LZW - Observações

Dicionário de tamanho limitado a P frases

Número de bits, b, dos índices das frases pode crescer conforme necessário, a medida que novas frases são inseridas

No LZW original, índices de tamanho fixo
 $b = \log_2 P$

LZW - Observações

O que fazer quando o dicionário enche?

Eliminar frases usadas menos recentemente (LRU)

Eliminar frases menos utilizadas (LFU)

Manter dicionário estático

Reiniciar dicionário

Monitorar compressão e, caso se detecte redução na RC, reiniciar dicionário.

Burrows-Wheeler Transform (BWT)

Transformada reversível criada por Wheeler in 1983, mas não publicada

Aplicação para compressão: Michael Burrows and David Wheeler, 1994

A BWT não comprime, mas torna a sequência mais apropriada à compressão por métodos simples

Burrows-Wheeler Transform (BWT)

Mensagem com N símbolos

Passos:

1. Listar as N mensagens diferentes obtidas por rotações (deslocamentos cílicos) da mensagem original
2. Ordenar lexicograficamente as N mensagens obtidas com as rotações e indexar a lista ordenada
3. Reter o índice da mensagem original e o último símbolo das mensagens da lista ordenada, preservando a ordem

WT

Exemplo:

$W = \text{quequalquasequando}$ ($N = 18$)

Alfabeto = {a, d, e, l, n, o, q, s, u}

Rotações:

quequalquasequando

oquequalquasequand

doquequalquasequan

...

BWT – Rotações

quequalquasequando
o queequalquasequand
do queequalquasequau
ndo queequalquasequa
ando queequalquasequ
uando queequalquaseq
quando queequalquase
equando queequalquas
sequando queequalqua
ase quando queequalqu
uase quando queequalq
quase quando queequal
l quase quando quequa
al quase quando quequ
ual quase quando queq
qual quase quando que
equal quase quando qu
ue qual quase quando q

BWT – Ordenação e Indexação

0 alquasequandoquequ
1 andoquequalquasequ
2 asequandoqueequalqu
3 doqueequalquasequan
4 equalquasequandoqu
5 equandoqueequalquas
6 lquasequandoquequa
7 ndoqueequalquasequa
8 oqueequalquasequand
9 qualquasequandoque
10 quandoqueequalquase
11 quasequandoquequal
12 quequalquasequando
13 sequandoqueequalqua
14 ualquasequandoqueq
15 uandoqueequalquaseq
16 uasequandoqueequalq
17 uequalquasequandoq

BWT – Passo 3

Mensagem transformada:

12 uuunusaadeeloaqqqq

BWT – Passo 3

A BWT informa explicitamente o último símbolo de cada rotação na lista ordenada

Lista ordenada \Rightarrow sabe-se o símbolo que se segue a cada um destes símbolos finais, na mensagem original

BWT – Passo 3

O 1º símbolo da 1^a rotação ('a') encontra-se na última posição da rotação de índice 6, e é o símbolo que se segue ao último símbolo da 1^a rotação ('u')

O 1º símbolo da 2^a rotação ('a') encontra-se na última posição da rotação de índice 7, e é o símbolo que se segue ao último símbolo da 2^a rotação ('u')

BWT – Passo 3

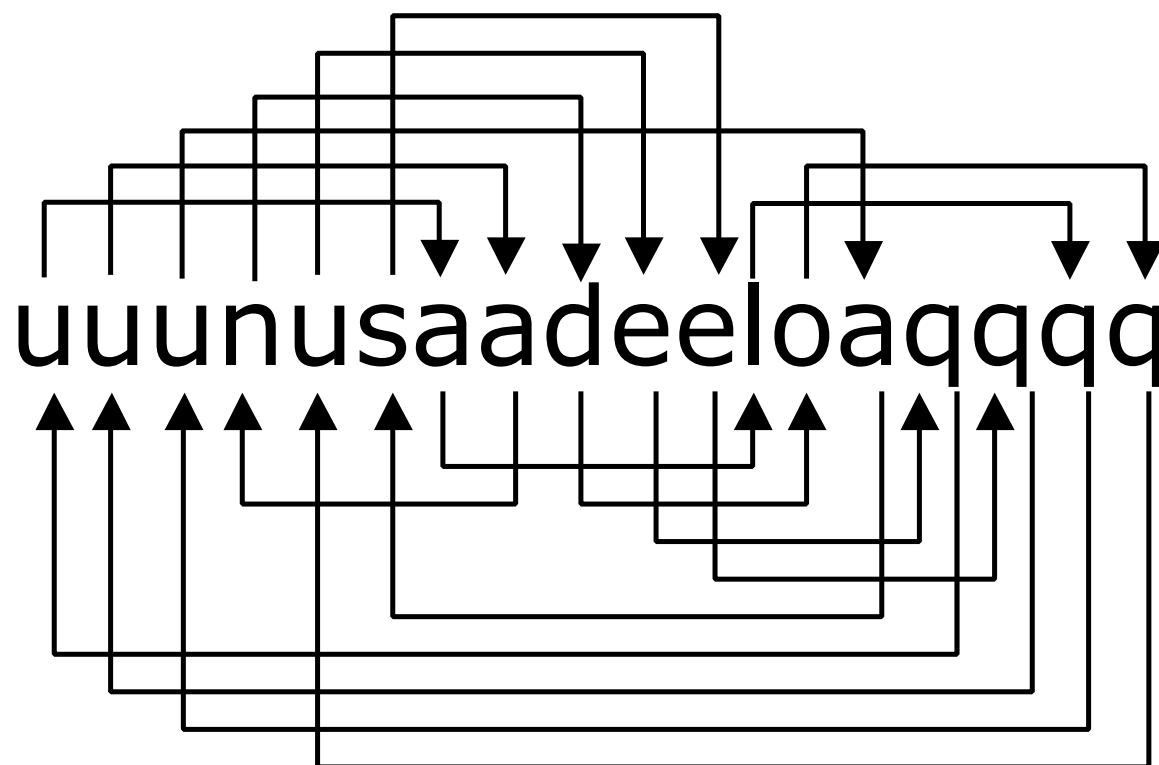
O 1º símbolo da 3^a rotação ('a') encontra-se na última posição da rotação de índice 13, e é o símbolo que se segue ao último símbolo da 3^a rotação ('u')

O 1º símbolo da 4^a rotação ('d') encontra-se na última posição da rotação de índice 8, e é o símbolo que se segue ao último símbolo da 4^a rotação ('n')

...

BWT – Passo 3

Localizam-se os símbolos seguintes observando-se apenas a BWT:



BWT – Passo 3

Índice do símbolo na BWT \Rightarrow índice do símbolo seguinte na mensagem original:

u	u	u	n	u	s	a	a	d	e	e	1	o	a	q	q	q	q
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
6	7	13	8	9	10	11	3	12	14	15	16	17	5	0	1	2	4

BWT – Decodificação

Índice da mensagem original informa a posição na BWT do último símbolo da mensagem

Mapeamento informa o índice do símbolo seguinte, ou seja, do 1º símbolo da mensagem

Seguindo o mapeamento encontram-se todos os símbolos seguintes

BWT – Decodificação

12 \Rightarrow 17 (q) \Rightarrow 4 (u) \Rightarrow 9(e) \Rightarrow 14(q) \Rightarrow 0(u)
 \Rightarrow 6(a) \Rightarrow 11(l) \Rightarrow 16(q) \Rightarrow 2(u) \Rightarrow 13(a) \Rightarrow
5(s) \Rightarrow 10(e) \Rightarrow 15(q) \Rightarrow 1(u) \Rightarrow 7(a) \Rightarrow 3(n)
 \Rightarrow 8(d) \Rightarrow 12(o)

u	u	u	n	u	s	a	a	d	e	e	l	o	a	q	q	q	q
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
6	7	13	8	9	10	11	3	12	14	15	16	17	5	0	1	2	4

Compressão com BWT

Passos:

BWT

Move-To-Front

Codificação de entropia (aritmético)

Move-To-Front (MTF)

Inicialmente, símbolos são ordenados e indexados (índices 0, 1, ...)

Na codificação, cada símbolo da mensagem é substituído pelo seu índice, e então passa a ocupar a 1^a posição na lista

MTF - exemplo

$W = \text{quequalquasequando}$

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	a	d	e	l	n	o	q	s	u

Codificação do 1º símbolo da BWT: 8

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	u	a	d	e	l	n	o	q	s

Codificação do 2º símbolo: 0

Codificação do 3º símbolo: 0

Codificação do 4º símbolo: 5

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	n	u	a	d	e	l	o	q	s

Codificação do 5º símbolo: 1

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	u	n	a	d	e	l	o	q	s

Codificação do 6º símbolo: 8

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	s	u	n	a	d	e	l	o	q

Codificação do 7º símbolo: 3

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	a	s	u	n	d	e	l	o	q

Codificação do 8º símbolo: 0

Codificação do 9º símbolo: 4

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	d	a	s	u	n	e	l	o	q

Codificação do 10º símbolo: 5

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	e	d	a	s	u	n	l	o	q

Codificação do 11º símbolo: 0

Codificação do 12º símbolo: 6

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	l	e	d	a	s	u	n	o	q

Codificação do 13º símbolo: 7

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	o	l	e	d	a	s	u	n	q

Codificação do 14º símbolo: 4

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	a	o	l	e	d	s	u	n	q

Codificação do 15º símbolo: 8

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	q	a	o	l	e	d	s	u	n

Codificação do 16º símbolo: 0

Codificação do 17º símbolo: 0

Codificação do 18º símbolo: 0

MTF - exemplo

$\text{MTF}(\text{BWT}(W)) =$

8 0 0 5 1 8 3 0 4 5 0 6 7 4 8 0 0 0

Huffman semi-adaptativo não-contextual: 53 bits sobre W , e 48 bits sobre $\text{MTF}(\text{BWT}(W))$

Ganho de compressão tende a aumentar com o aumento de W

Modelos não-contextuais são eficientes com a BWT, que já explora informação contextual

Modelos Determinísticos

Modelos estatísticos: número de contextos diferentes cresce exponencialmente com o tamanho do contexto

Alternativa: modelos determinísticos de predição

Sinais físicos: predição polinomial de ordem K

Predição Polinomial de Ordem K

Predição do próximo símbolo baseada nos últimos K símbolos

Mensagem: $W = w_0w_1\dots w_{i-1}$

Predição para o próximo símbolo: v_i

Valor real do próximo símbolo: w_i

Erro de predição: $e_i = w_i - v_i$

Predição Polinomial de Ordem K

K	v_i	e_i
1	w_{i-1}	$w_i - w_{i-1}$
2	$2w_{i-1} - w_{i-2}$	$w_i - 2w_{i-1} + w_{i-2}$
3	$3w_{i-1} - 3w_{i-2} + w_{i-3}$	$w_i - 3w_{i-1} + 3w_{i-2} - w_{i-3}$

Predição Polinomial de Ordem K

Exemplo: fonte octal

$W = 7\ 6\ 5\ 4\ 4\ 4\ 3\ 4\ 3\ 2\ 1\ 0\ 0\ 0\ 0\ 1\ 2\ 3\ 2\ 1$
 $1\ 2\ 3\ 5\ 5\ 5\ 6\ 6\ 6\ 7\ 7\ 7\ (N = 32 \text{ símbolos})$

$N_0 = N_1 = N_2 = \dots = N_7$

$H = 3$ bits/símbolo = comprimento médio
do código de Huffman semiadaptativo
não contextual

Predição Polinomial de Ordem K

$K = 1$

$e = 7 \ -1 \ -1 \ -1 \ 0 \ 0 \ -1 \ 1 \ 1 \ -1 \ -1 \ -1 \ -1 \ 0 \ 0 \ 0 \ 1$
 $1 \ 1 \ -1 \ -1 \ 0 \ 1 \ 1 \ 2 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0$

Desconsiderando o símbolo inicial:

$N_0=12; N_{-1}=10; N_1=8; N_2=1$

Modelo semiadaptativo não-contextual:

$H = 1,72$ bits/símbolo

Compr. médio de Huffman = 1,9
bits/símbolo

Predição Polinomial de Ordem K

$$K = 2$$

$$V = 7 \ 6 \ 0 \ 0 \ 1 \ 0 \ -1 \ 2 \ -2 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ - \\ 2 \ 0 \ 1 \ 1 \ 0 \ 1 \ -2 \ 0 \ 1 \ -1 \ 0 \ 1 \ -1 \ 0$$

Desconsiderando os dois símbolos iniciais:

$$N_0=15; N_1=8; N_{-1}=3; N_{-2}=3; N_2=1$$

$$H = ? \text{ bits/símbolo}$$

Compr. médio código de Huffman = 1,8 bits/símbolo

Predição Polinomial de Ordem K

$K = 3$

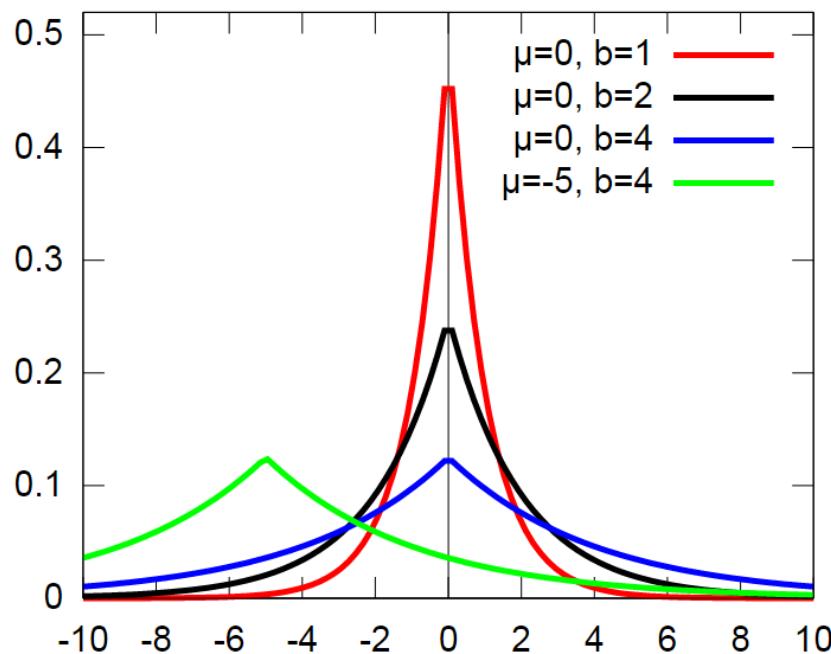
$V = 7 \ 6 \ 5 \ 4 \ 3 \ 5 \dots$

$H = ?$

Compr. médio código de Huffman = ?

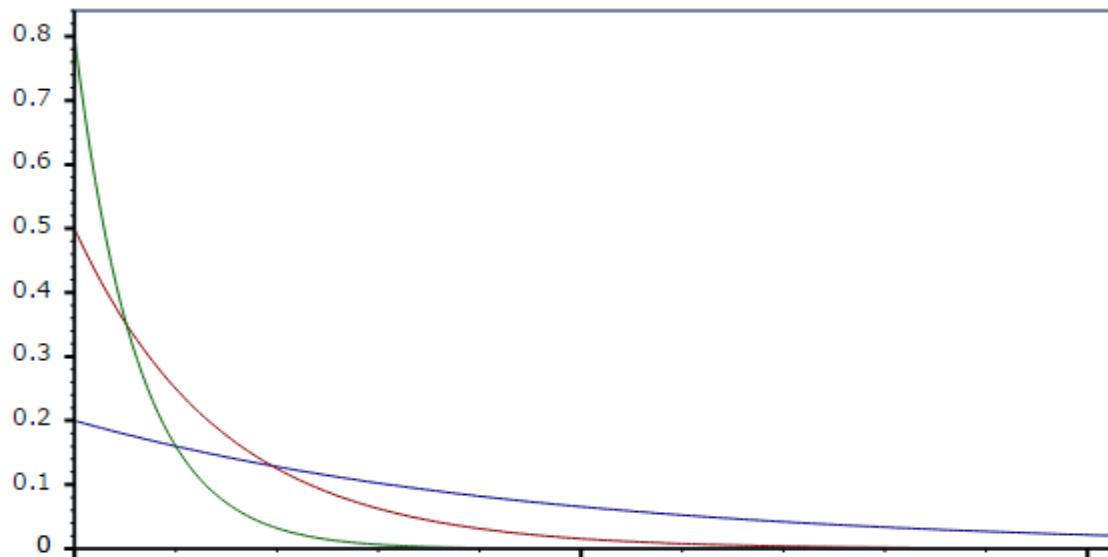
Predição Polinomial de Ordem K

Para sinais físicos, erros de predição tendem a apresentar uma distribuição laplaciana (dupla exponencial) de média 0



Predição Polinomial de Ordem K

Um mapeamento adequado (valores negativos para positivos ímpares e valores positivos para positivos pares), gera uma aproximação para uma distribuição geométrica.



Predição Polinomial de Ordem K

Codificadores extremamente simples e eficazes para distribuições geométricas:

Golomb, ótimo de comprimento inteiro

Golomb-Rice, subótimo

Compressão com perdas

Distorções introduzidas pela compressão: aceitáveis para vídeo, imagens, som etc.

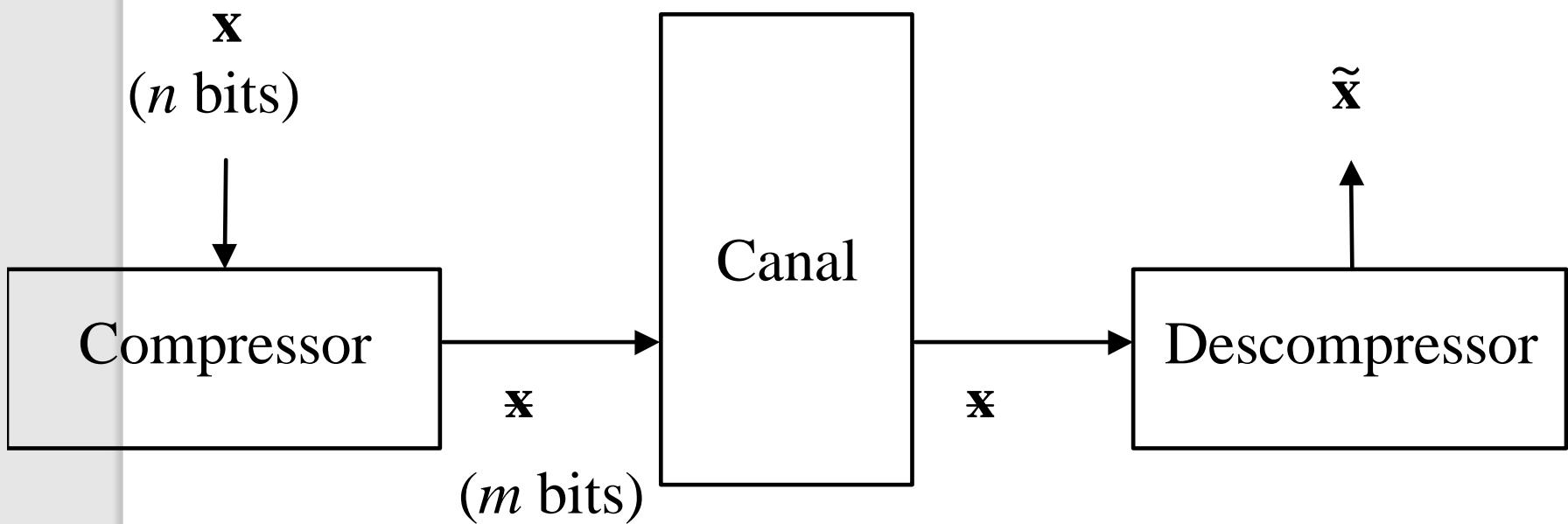
Sinais críticos (e. g. utilizados no diagnóstico médico): distorções cuidadosamente controladas

Compromisso RC x distorção

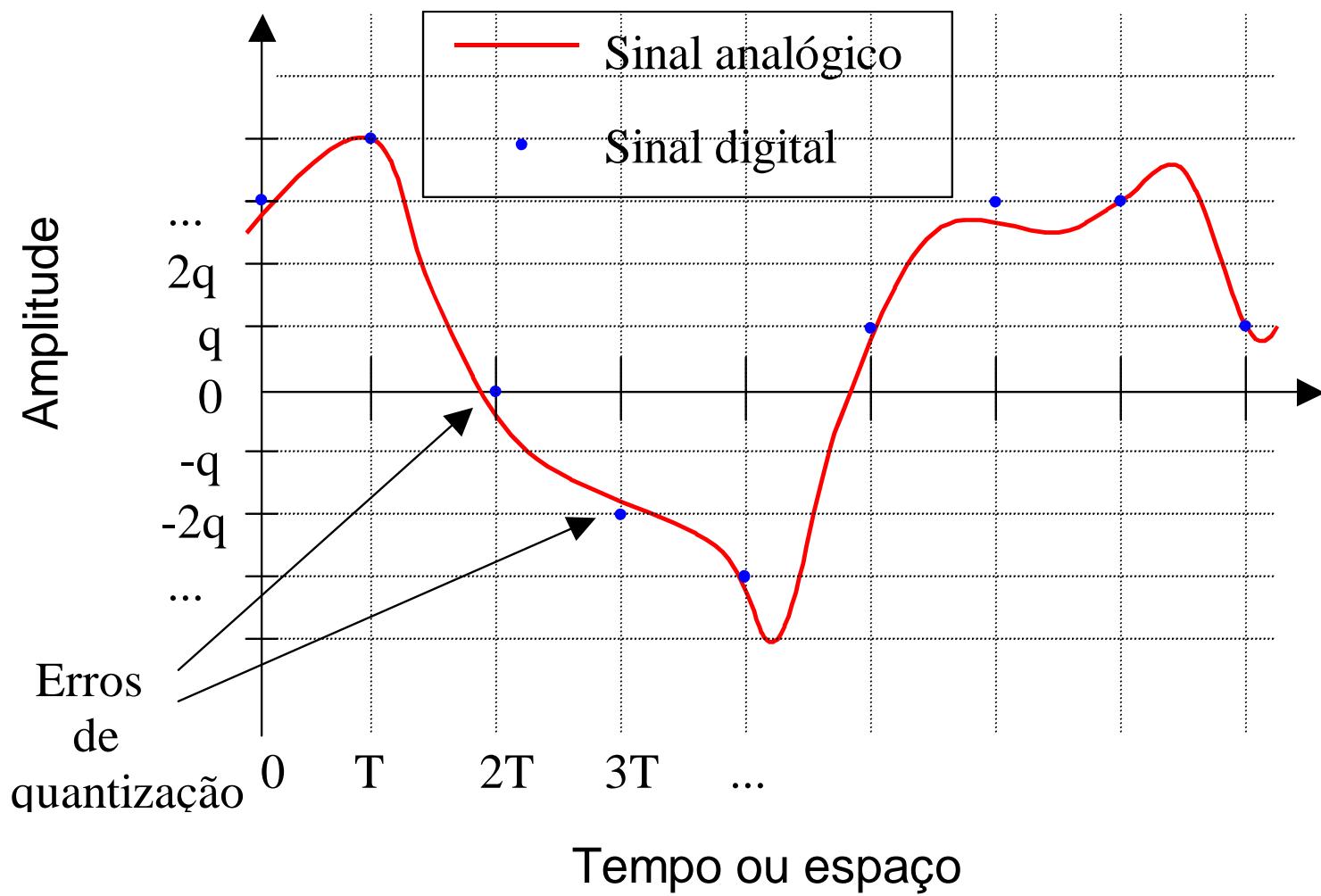
Subamostragem ou quantização

Compressão com perdas

Compressor/descompressor genérico



Amostragem e Quantização



Quantização Vetorial (QV)

Complexidade computacional elevada para compressão

Complexidade computacional reduzida para descompressão

Shannon, 1948: codificar elementos agrupados é mais eficaz que isoladamente.

Quantização Vetorial (QV)

K vetores-código (sequências de L símbolos da fonte) selecionados são armazenados em um *codebook* disponível para o compressor e para o descompressor

Vetores-código devem representar a fonte

Mensagens particionadas em blocos de L elementos consecutivos, substituídos por vetores semelhantes do codebook.

Quantização Vetorial (QV)

Ex: fonte octal, $L = 3$, mensagem
 w

$$w = \left| \begin{array}{c} 7 \ 6 \ 5 \ | 4 \ 4 \ 4 \ | 3 \ 4 \ 3 \ | 2 \ 1 \ 0 \ | 0 \ 0 \ 0 \ | 1 \ 2 \ 3 \ | 2 \ 1 \ 0 \ | \dots \\ \mathbf{v}_0 \qquad \mathbf{v}_1 \qquad \mathbf{v}_2 \qquad \mathbf{v}_3 \qquad \mathbf{v}_4 \qquad \mathbf{v}_5 \qquad \mathbf{v}_6 \qquad \dots \end{array} \right|$$

Quantização Vetorial (QV)

Alfabeto original com cardinalidade M:
 $V = M^L$ vetores diferentes

Exemplos:

$$M = 8 \text{ e } L = 3 \Rightarrow V = 8^3 = 512$$

$$M = 16 \text{ e } L = 4 \Rightarrow V = 16^4 = 64K$$

$$M = 256 \text{ e } L = 4 \Rightarrow V = 256^4 = 4G$$

Quantização Vetorial (QV)

\mathbf{c}_k , $k = 0, 1, \dots, K-1$: k -ésimo vetor-código do *codebook*

c_{kj} , $j = 0, 1, \dots, L-1$: j -ésimo elemento de \mathbf{c}_k

\mathbf{v}_i , $i = 0, 1, \dots$: i -ésimo vetor na mensagem produzida pela fonte

v_{ij} , $j = 0, 1, \dots, L-1$: j -ésimo elemento de \mathbf{v}_i

Quantização Vetorial (QV)

Um codebook exemplo

k	c_{k0}	c_{k1}	...	$c_{k,L-1}$
0	100	110	...	99
1	45	225	...	230
...
$K-1$	2	0	...	3

Quantização Vetorial (QV)

Ex: fonte octal, L=3, mensagem W, K=16

Codebook
exemplo:

k	Vetor \mathbf{c}_k		
0	0	1	0
1	7	6	6
2	3	0	5
...
14	3	3	3
15	1	1	2

Quantização Vetorial (QV)

Codificar \mathbf{v}_i ; distorção $D(\mathbf{v}_i, \mathbf{c}_k)$, $k = 0, 1, \dots, K-1$

Erro Médio Quadrático (EMQ):

$$D(\mathbf{v}_i, \mathbf{c}_k) = \frac{1}{L} \sum_{j=0}^{L-1} (v_{ij} - c_{kj})^2$$

$D(\mathbf{v}_i, \mathbf{c}_k)$ mínima para $k = k'$: índice k' anexado à mensagem comprimida

Descompressão: k' substituído por $\mathbf{c}_{k'}$

Quantização Vetorial (QV)

Ex: fonte octal, $L = 3$, mensagem W

$$W = 7 \ 6 \ 5 \ 4 \ 4 \ 4 \ 3 \ 4 \ 3 \ 2 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 2 \ 1 \ 0 \ \dots$$
$$\mathbf{v}_0 \quad \mathbf{v}_1 \quad \mathbf{v}_2 \quad \mathbf{v}_3 \quad \mathbf{v}_4 \quad \mathbf{v}_5 \quad \mathbf{v}_6 \quad \dots$$

k Vetor \mathbf{c}_k

0	0	1	0
1	7	6	6
2	3	0	5
...
14	3	3	3
15	1	1	2

$$D(\mathbf{v}_0, \mathbf{c}_0) = (7-0)^2 + (6-1)^2 + (5-0)^2 = 99$$

$$D(\mathbf{v}_0, \mathbf{c}_1) = (7-7)^2 + (6-6)^2 + (5-6)^2 = 1$$

...

$$D(\mathbf{v}_0, \mathbf{c}_{63}) = (7-1)^2 + (6-1)^2 + (5-2)^2 = 70$$

Se $D(\mathbf{v}_0, \mathbf{c}_1)$ é o menor valor, \mathbf{v}_0 é codificado como 1, com $\log_2 16 = 4$ bits $\Rightarrow RC = 9/4$.

Quantização Vetorial (QV)

Se os índices são codificados com $\log_2(K)$ bits, a taxa resultante é de $\log_2(K)/L$ bits/elemento

Alfabeto original com cardinalidade M:

$$RC = L \times \log_2(M)/\log_2(K)$$

Ex: M = 256, L = 4, K = 128

$$RC = 4\log_2(256)/\log_2(128) = 4,57$$

Quantização Vetorial (QV)

Teoricamente, quanto maior L , mais eficaz é a quantização

Dimensão dos vetores limitada na prática

Desempenho depende ainda do tamanho do *codebook*, da escolha dos vetores-código e da medida de distorção

Como selecionar vetores-código?

Quantização Vetorial (QV)

O *codebook* é gerado por *treinamento*:

Longas mensagens examinadas

Vetores representativos selecionados como vetores-código: Algoritmo LBG, redes neurais etc

Codebooks estáticos x dinâmicos

Quantização Vetorial (QV)

Exemplo:

$W = \begin{matrix} 7 & 6 & 5 & 4 & 4 & 4 & 3 & 4 & 3 \\ 2 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 0 & 0 & 1 & 2 & 3 & 5 \\ 5 & 5 & 6 & 6 & 6 & 7 & 7 & 7 & 5 \end{matrix}$

$L = 3, K = 8$

Compressão sem e
com Huffman adaptativo
não contextual

EMQ? Erro máximo?

k	c_k		
0	0	0	0
1	1	2	3
2	3	1	1
3	4	4	3
4	6	0	0
5	5	3	7
6	6	3	7
7	7	7	6

Quantização Escalar

Redução de alfabeto, símbolos tratados individualmente

Quantização uniforme: divisão pelo passo de quantização, q , seguida por arredondamento para inteiro

Valor “quantizado”: $\hat{x} = x // q$

Valor “dequantizado” $\tilde{x} = q\hat{x}$

Quantização Escalar

Ex: Alfabeto $A = \{0, 1, \dots, 255\}$

$$q = 2 \Rightarrow Aq_2 = \{0, 1, \dots, 128\}$$

$$Ad_2 = \{0, 2, 4, \dots, 256\}$$

$$q = 3 \Rightarrow Aq_3 = \{0, 1, \dots, 85\}$$

$$Ad_3 = \{0, 3, 6, \dots, 255\}$$

$$q = 4 \Rightarrow Aq_4 = \{0, 1, \dots, 64\}$$

$$Ad_4 = \{0, 4, 8, \dots, 256\}$$

$$E_{\max} = \lfloor q/2 \rfloor$$

Paradigma Transformada, Quantização, Codificação (TQC)

Transformadas: Cosseno Discreta
(DCT) e wavelets, principalmente

Quantização: normalmente escalar

Codificação de entropia normalmente
simples (até estática!)

Paradigma TQC

Baixo custo computacional, compromisso
Taxa-Distorção próximo a VQ

Utilizado nos padrões JPEG, JPEG2000,
MPEG, H265 (MPEG-H) , MP3, AAC...

JPEG - Transformada

Banda_X = Banda_X-128

Aplicação da DCT bidimensional a blocos
(8 x 8 pixels) adjacentes da Banda_X.

Transformada Cosseno Discreta

DCT de $x[n]$, $n = 0, 1 \dots N-1$

$$X[k] = \left(\frac{2}{N} \right)^{1/2} c_k \sum_{n=0}^{N-1} x[n] \cos \left[2\pi \frac{k}{2N} n + \frac{k\pi}{2N} \right], \quad k = 0, 1, \dots, N-1$$

DCT inversa (IDCT) de $X[k]$

$$x[n] = \left(\frac{2}{N} \right)^{1/2} \sum_{k=0}^{N-1} c_k X[k] \cos \left[2\pi \frac{k}{2N} n + \frac{k\pi}{2N} \right], \quad n = 0, 1, \dots, N-1$$

$$c_k = \begin{cases} (1/2)^{1/2} & \text{para } k = 0 \\ 1 & \text{para } k = 1, 2, \dots, N-1 \end{cases}$$

Transformada Cosseno Discreta

$X[k]$: coeficientes DCT

X : representação de x no domínio da frequência

$X[0]$: coeficiente DC (Direct Current)

$X[1] \dots X[N-1]$: coeficientes AC (Alternate Current)

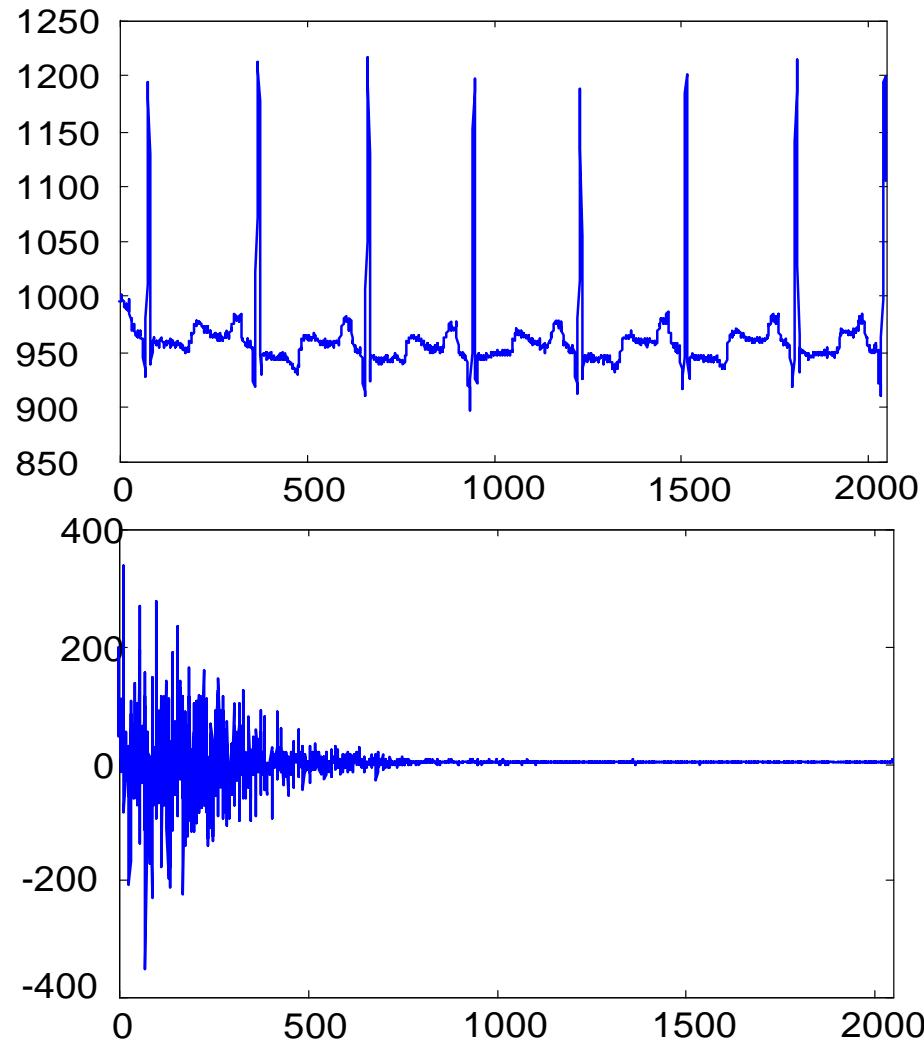
Complexidade computacional elevada

Algoritmos eficientes: FDCT

DCT – Exemplo

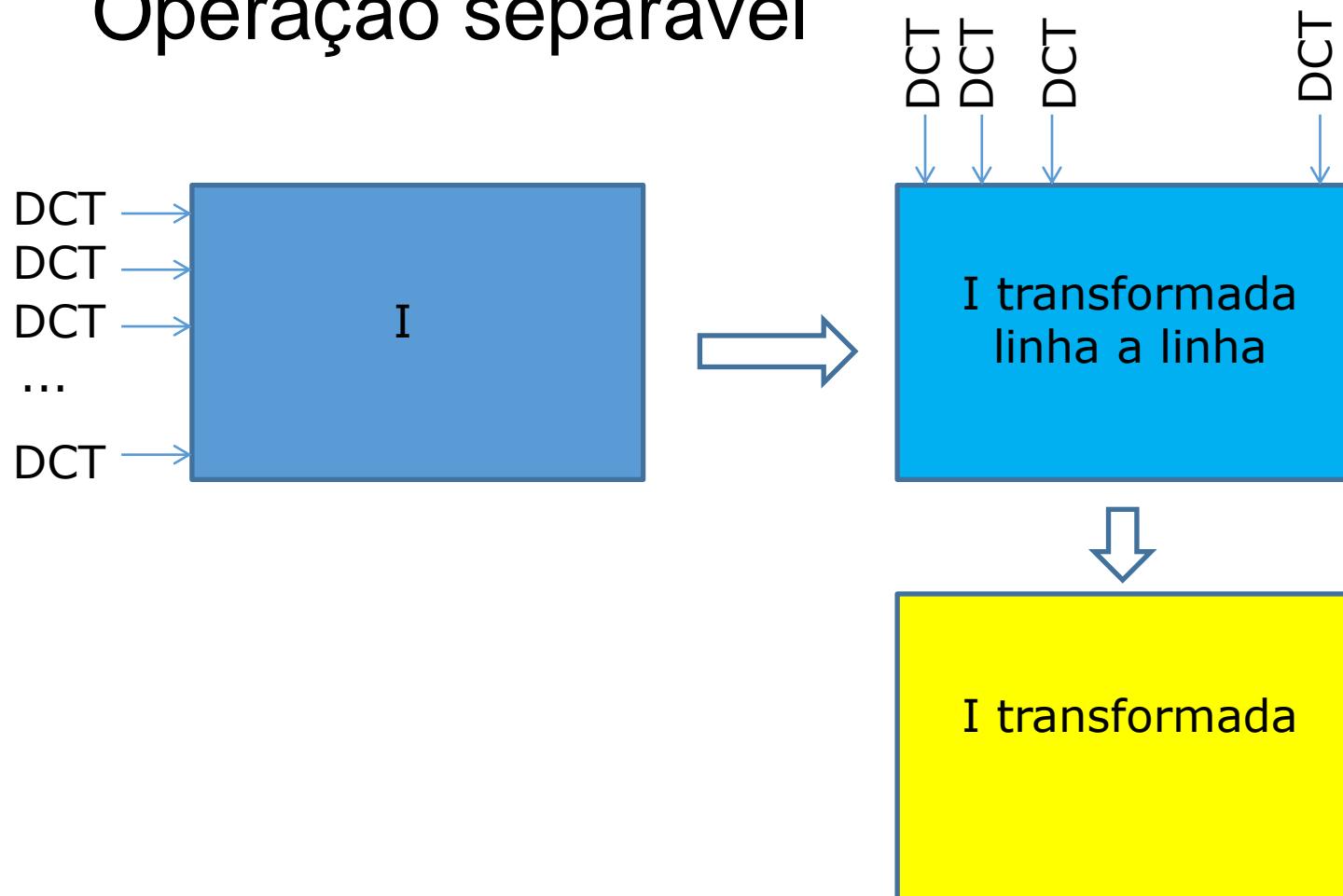
Sinal
eletrocardiográfico,
2048 amostras

DCT do sinal
eletrocardiográfico
(sem termo DC)

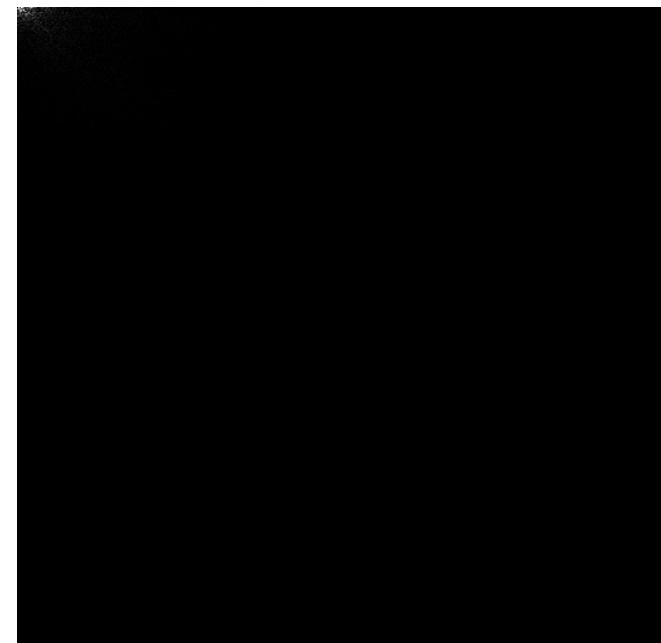


DCT 2-D

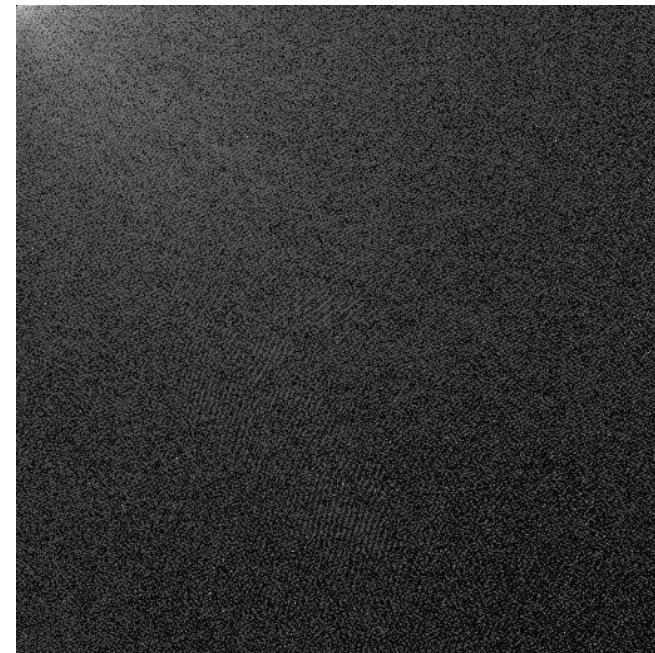
Operação separável



DCT 2-D - Exemplo



DCT 2-D - Exemplo



JPEG - Quantização

Matriz de quantização Q, 8 x 8

Q_{ij} é o passo de quantização

Quantização: BlocoDCT_{ij} // Q_{ij}

JPEG - Quantização

O comitê JPEG deixou livre a escolha de Q, mas divulgou matrizes de quantização obtidas por intermédio de experimentos psicofisiológicos.

JPEG - Quantização

Matriz de quantização sugerida

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	162
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

JPEG – Exemplo

Quantização/Dequantização

Bloco da Banda X

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

JPEG – Exemplo

Quantização/Dequantização

DCT de (Banda_X-128)

235,6	-1,0	-12,1	-5,2	2,1	-1,7	-2,7	1,3
-22,6	-17,5	-6,2	-3,2	-2,9	-0,1	0,4	-1,2
-10,9	-9,3	-1,6	1,5	0,2	-0,9	-0,6	-0,1
-7,1	-1,9	0,2	1,5	0,9	-0,1	0,0	0,3
-0,6	-0,8	1,5	1,6	-0,1	-0,7	0,6	1,3
1,8	-0,2	1,6	-0,3	-0,8	1,5	1,0	-1,0
-1,3	-0,4	-0,3	-1,5	-0,5	1,7	1,1	-0,8
-2,6	1,6	-3,8	-1,8	1,9	1,2	-0,6	-0,4

JPEG – Exemplo

Quantização/Dequantização

Q								
16	11	10	16	24	40	51	61	
12	12	14	19	26	58	60	55	
14	13	16	24	40	57	69	56	
14	17	22	29	51	87	80	162	
18	22	37	56	68	109	103	77	
24	35	55	64	81	104	113	92	
49	64	78	87	103	121	120	101	
72	92	95	98	112	100	103	99	

JPEG – Exemplo

Quantização/Dequantização

Bloco DCT quantizado por Q

JPEG – Exemplo

Quantização/Dequantização

Bloco DCT dequantizado

JPEG – Exemplo

Quantização/Dequantização

IDCT+128

144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	164	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

JPEG – Exemplo

Quantização/Dequantização

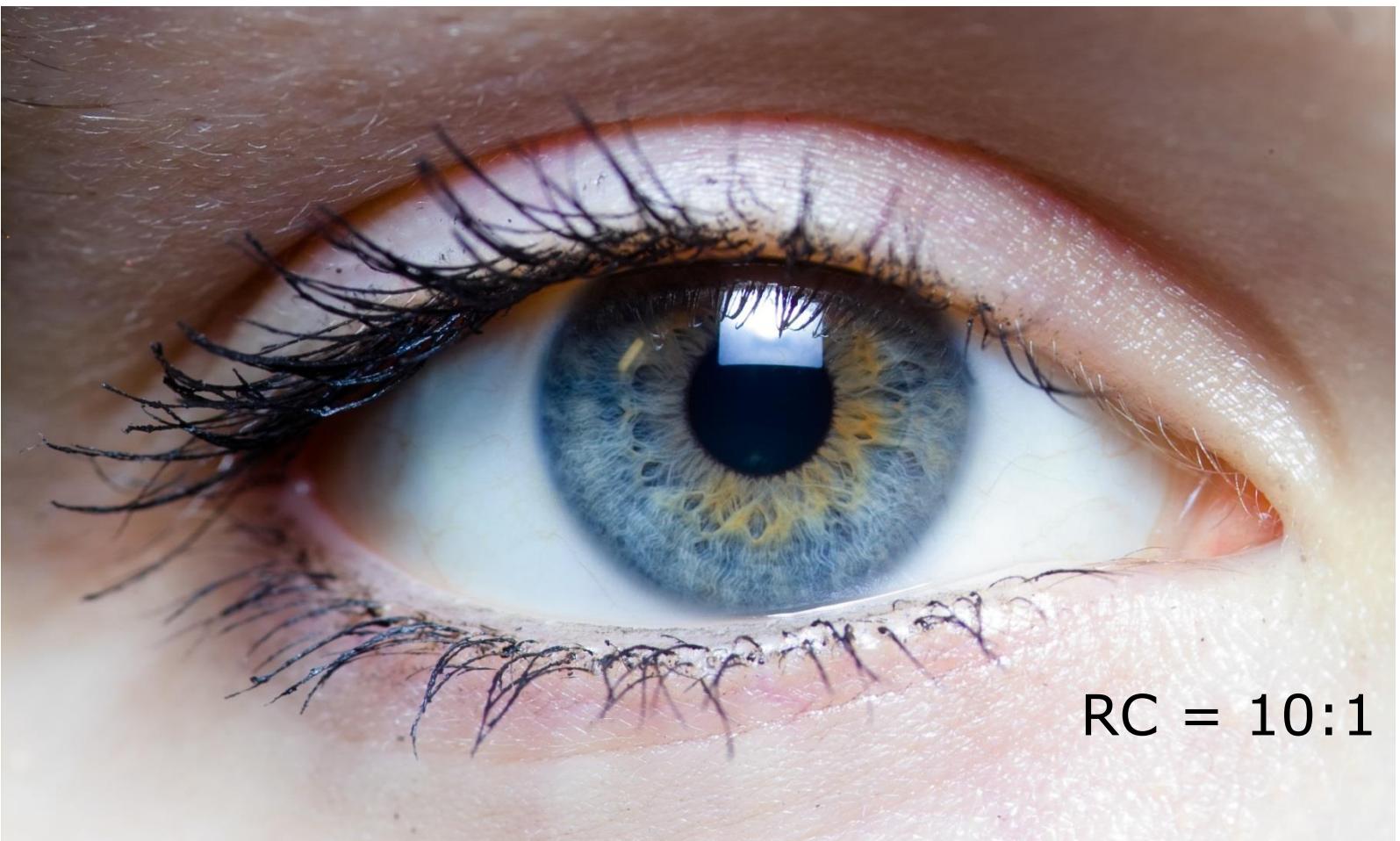
Original

139	144	149	153	155	155	155	155	144	146	149	152	154	156	156	156
144	151	153	156	159	156	156	156	148	150	152	154	156	156	156	156
150	155	160	163	158	156	156	156	155	156	157	158	158	157	156	155
159	161	162	160	160	159	159	159	160	161	161	162	161	159	157	155
159	160	161	162	162	155	155	155	163	163	164	163	162	160	158	156
161	161	161	161	160	157	157	157	163	164	164	164	162	160	158	157
162	162	161	163	162	157	157	157	160	161	162	162	162	161	159	158
162	162	161	161	163	158	158	158	158	159	161	161	162	161	159	158

Reconstrução

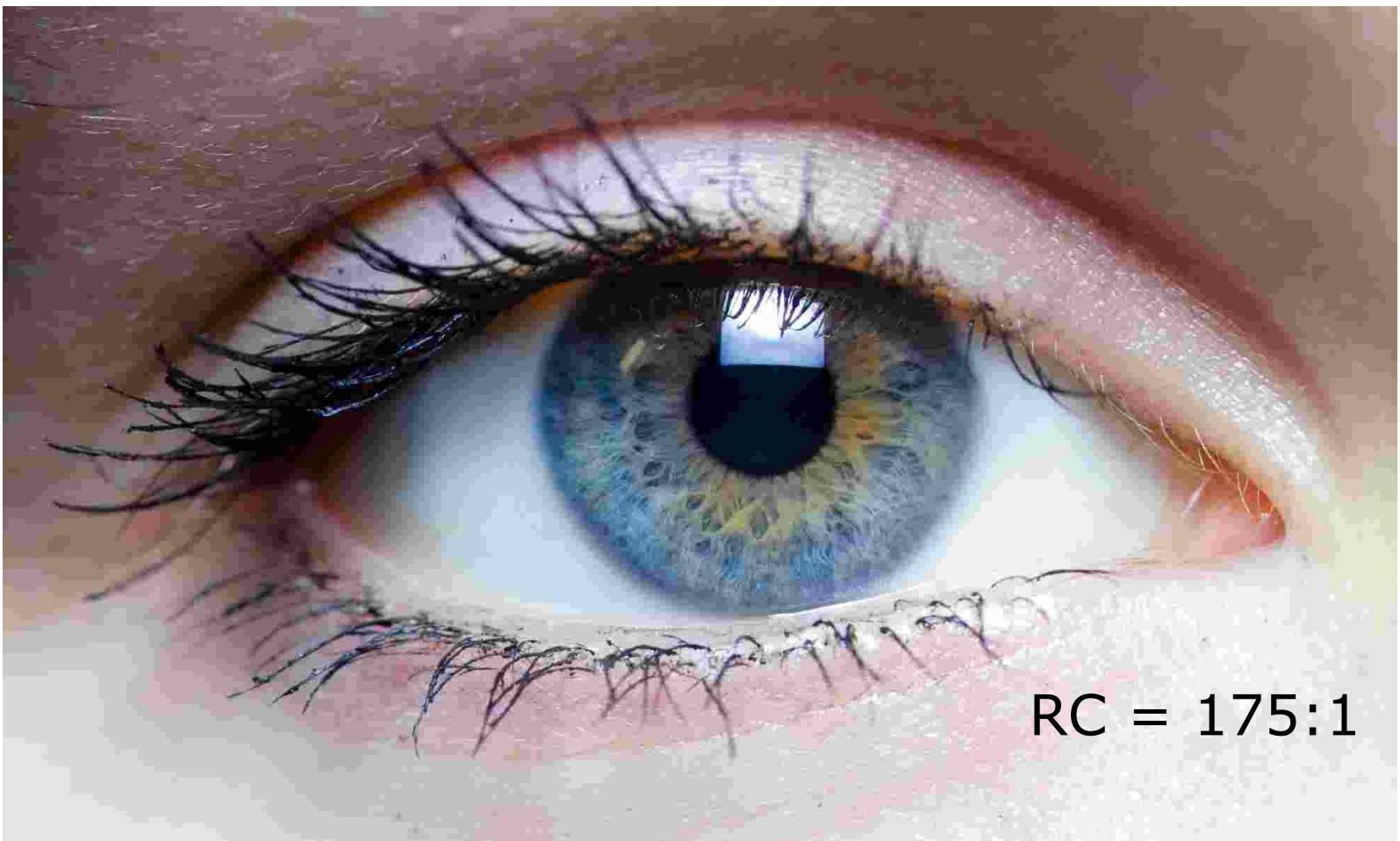
JPEG – Qualidade

Alta (1985 x 1194 x 24 bpp / 681KB)

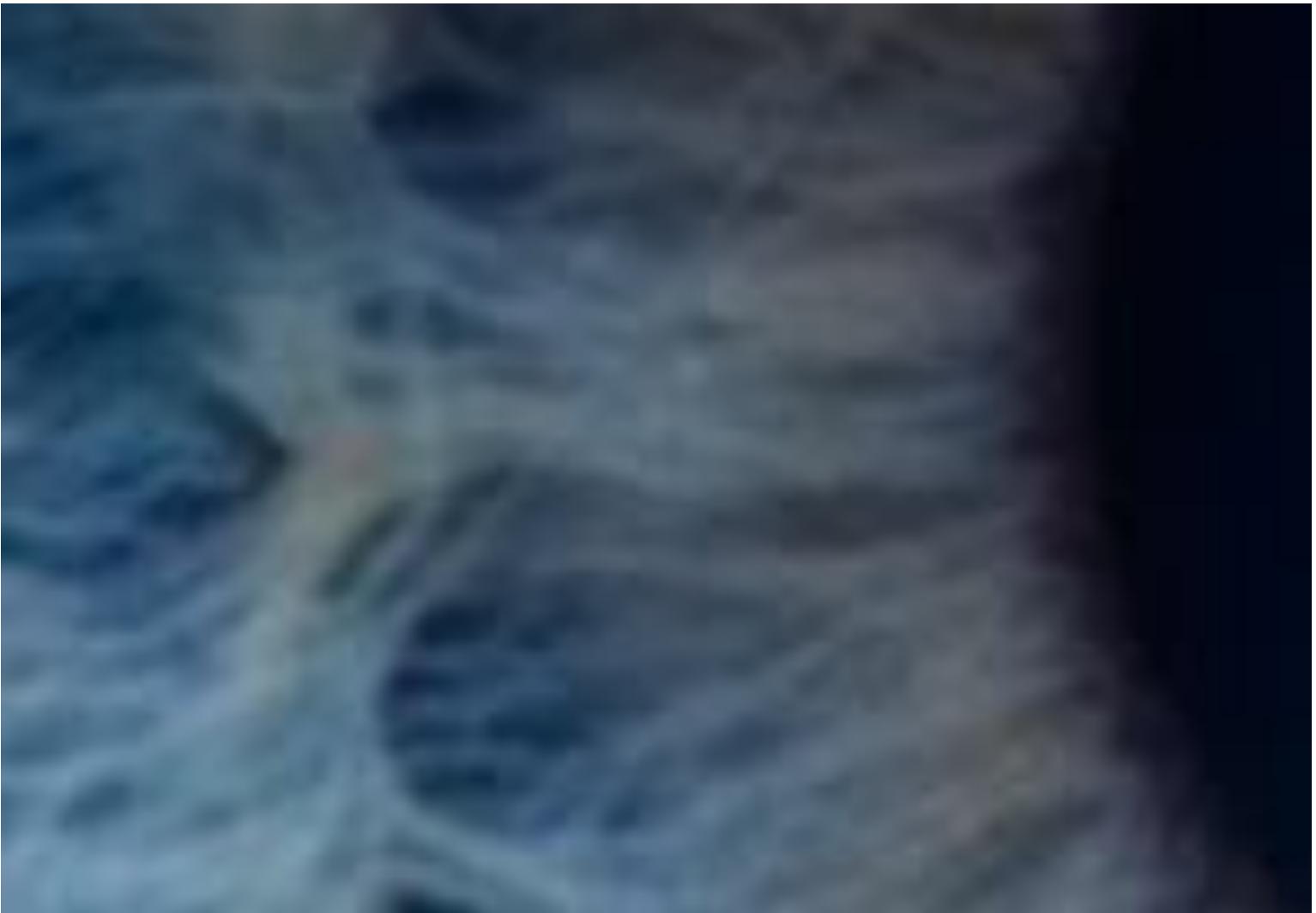


JPEG – Qualidade

Baixa (1985 x 1194 x 24 bpp / 39,7KB)



JPEG – Alta qualidade



JPEG – Baixa qualidade



JPEG – Codificação de Entropia

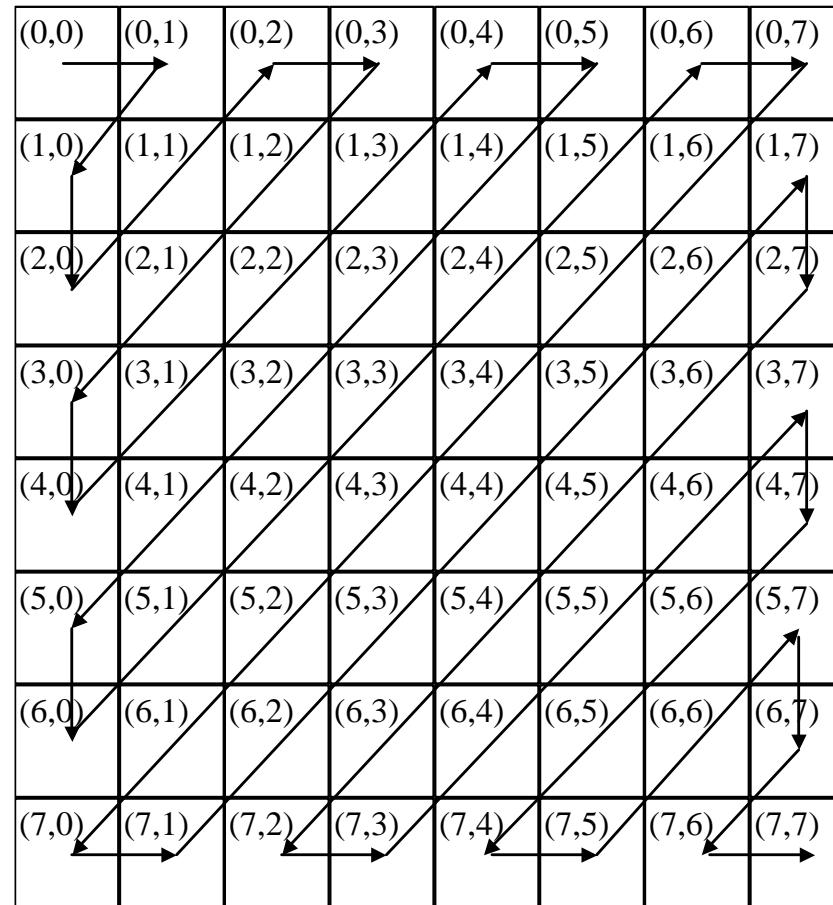
Predição de ordem 1 para os coeficientes DC de cada bloco

RLEZ dos coeficientes de cada bloco empregando percurso em zig-zag.

Codificação de Huffman ou Aritmética

JPEG – Codificação de Entropia

Percorso em zig-zag



JPEG – Codificação de Entropia

Coeficiente AC não nulo de valor a é representado em combinação com o tamanho c (≤ 15) da sequência de coeficientes AC nulos que o precede no percurso zig-zag

Cada par (c, a) é usualmente representado por um par de símbolos $s_1 = (c, b)$ e $s_2 = a$

b : nº de bits usados para codificar a

JPEG – Codificação de Entropia

$s_1 = (15, 0)$: sequência de 16 zeros, obrigatoriamente seguida por outro s_1 reiniciado para $c = 0$

$s_1 = (0,0)$: sequência de zeros de qualquer tamanho que se estende até o fim do bloco.

Coeficientes DC: $s_1 = b$ e $s_2 = a$ (valor do coeficiente DC diferencial).

JPEG – Codificação de Entropia

Relação entre b e a , para coeficientes AC:

b	a
1	-1, 1
2	-3, -2, 2, 3
3	-7, ..., -4, 4, ..., 7
4	-15, ..., -8, 8, ... 15
5	-31, ..., -16, 16, ..., 31
6	-63, ..., -32, 32, ..., 63
7	-127, ..., -64, 64, ..., 127
8	-255, ..., -128, 128, ..., 255
9	-511, ..., -256, 256, ..., 511
10	-1023, ..., -512, 512, ..., 1023

JPEG – Codificação de Entropia

Relação entre b e a , para coeficientes DC:

b	a
1	-1, 1
2	-3, -2, 2, 3
...	...
10	-1023, ..., -512, 512, ..., 1023
11	-2047,...-1024, 1024, 2047

JPEG – Codificação de Entropia

s_1 codificado por Huffman ou codificação aritmética, e s_2 por código fixo:

a	Código
-1	0
1	1
-3	00
-2	01
2	10
3	11
...	...

JPEG – Exemplo de Codificação de Entropia

Bloco DCT quantizado:

JPEG – Exemplo de Codificação de Entropia

Dados:

DC do bloco anterior: 12

Huffman para s_1 (DC) = 2: 011

Huffman para s_1 (AC):

s_1	Código
(0,0)	1010
(0,1)	00
(1,2)	11011
(2,1)	11100

JPEG – Exemplo de Codificação de Entropia

Coeficiente DC do bloco atual será substituído por $15-12 = 3$.

$a = 3$ será representado por $b = 2$ bits.

$$(s_1, s_2) = (2, 3)$$

Codificação: 011 11

JPEG – Exemplo de Codificação de Entropia

No percurso zig-zag,

-2 precedido por um 0:

$$(s_1, s_2) = [(1, 2), -2] \Rightarrow 11011\ 01$$

Três -1 adjacentes:

$$\text{três pares } (s_1, s_2) = [(0, 1), -1] \Rightarrow 00\ 0$$

Dois 0 seguido por -1:

$$(s_1, s_2) = [(2, 1), -1] \Rightarrow 11100\ 0$$

Zeros até o fim do bloco:

$$s_1 = (0, 0) \Rightarrow 1010$$

JPEG – Exemplo de Codificação de Entropia

Bloco DCT codificado:

011 11 11011 01 00 0 00 0 00 0 11100 0 1010

Se a imagem original usava 8 bits por pixel por banda, RC resultante $64 \times 8 / 31 = 16,5$.

MPEG

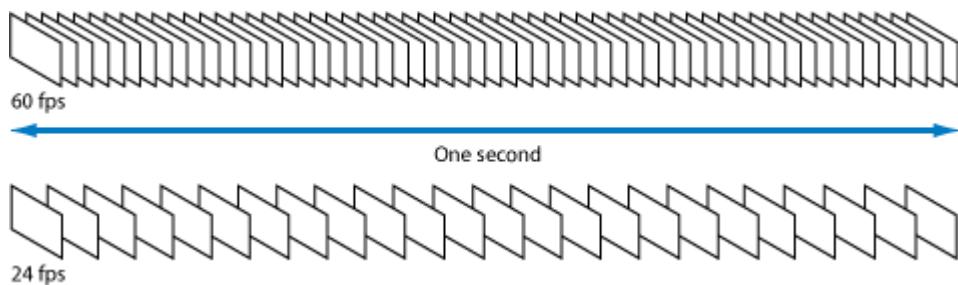
Moving Picture Experts Group

Sequência de quadros (imagens)

Remoção de redundância temporal e
espacial:

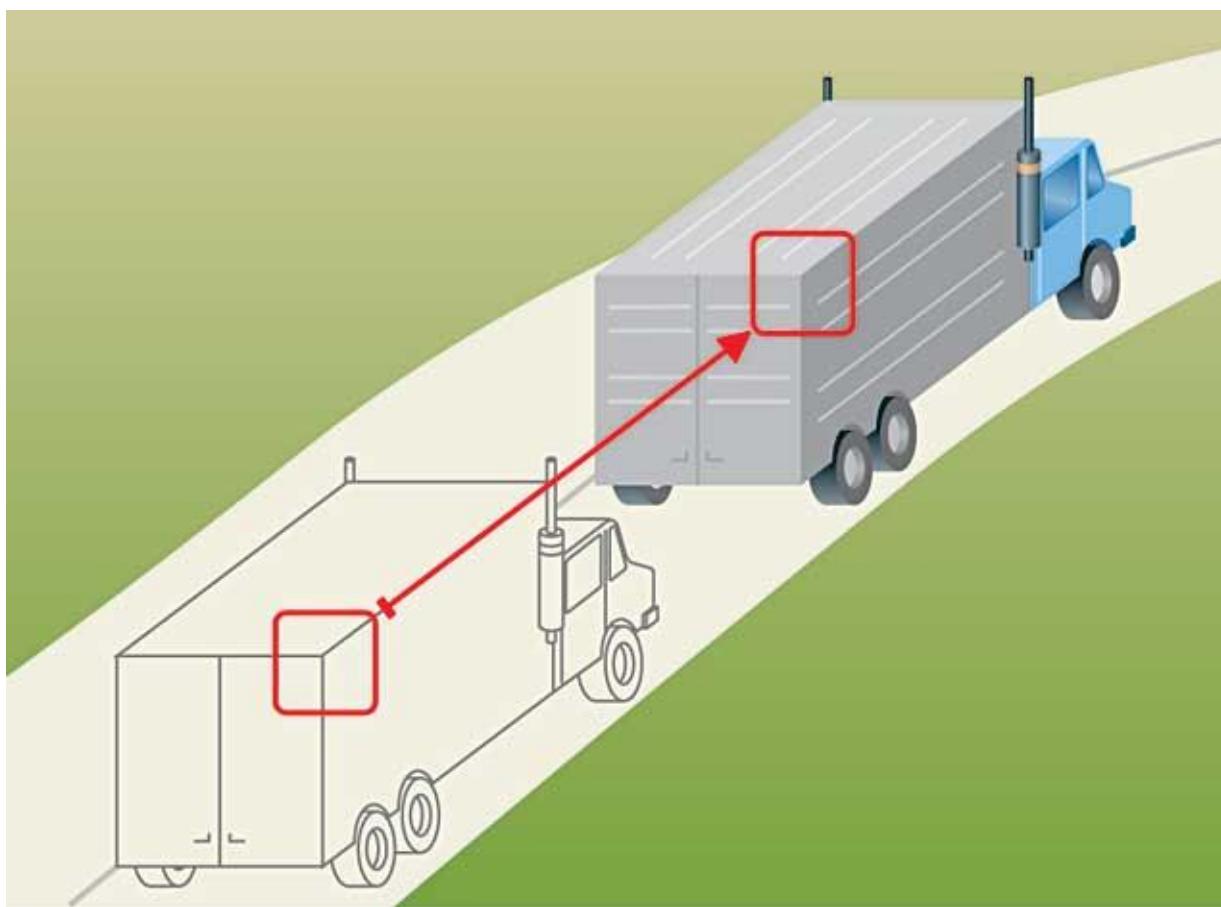
Estimação e compensação de movimento
de blocos interquadros + TQC

MPEG



MPEG

Estimação e compensação de movimento de blocos interquadros

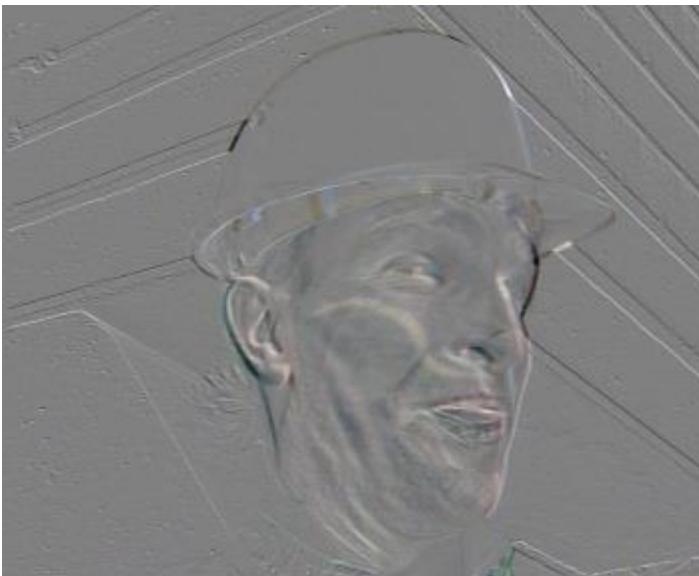


MPEG



MPEG

Sem compensação



Com compensação



MPEG



MPEG

Sem compensação

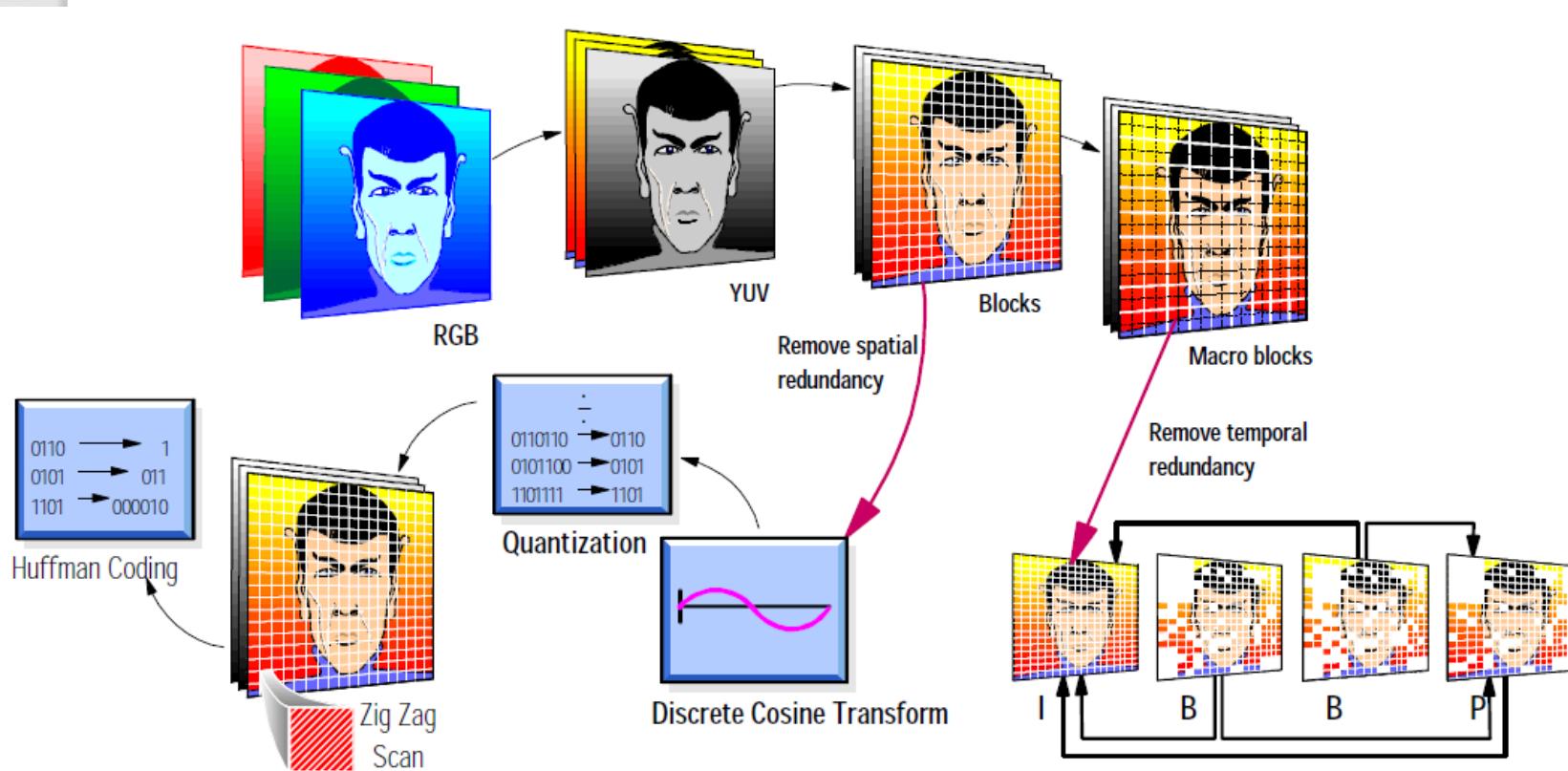


Com compensação



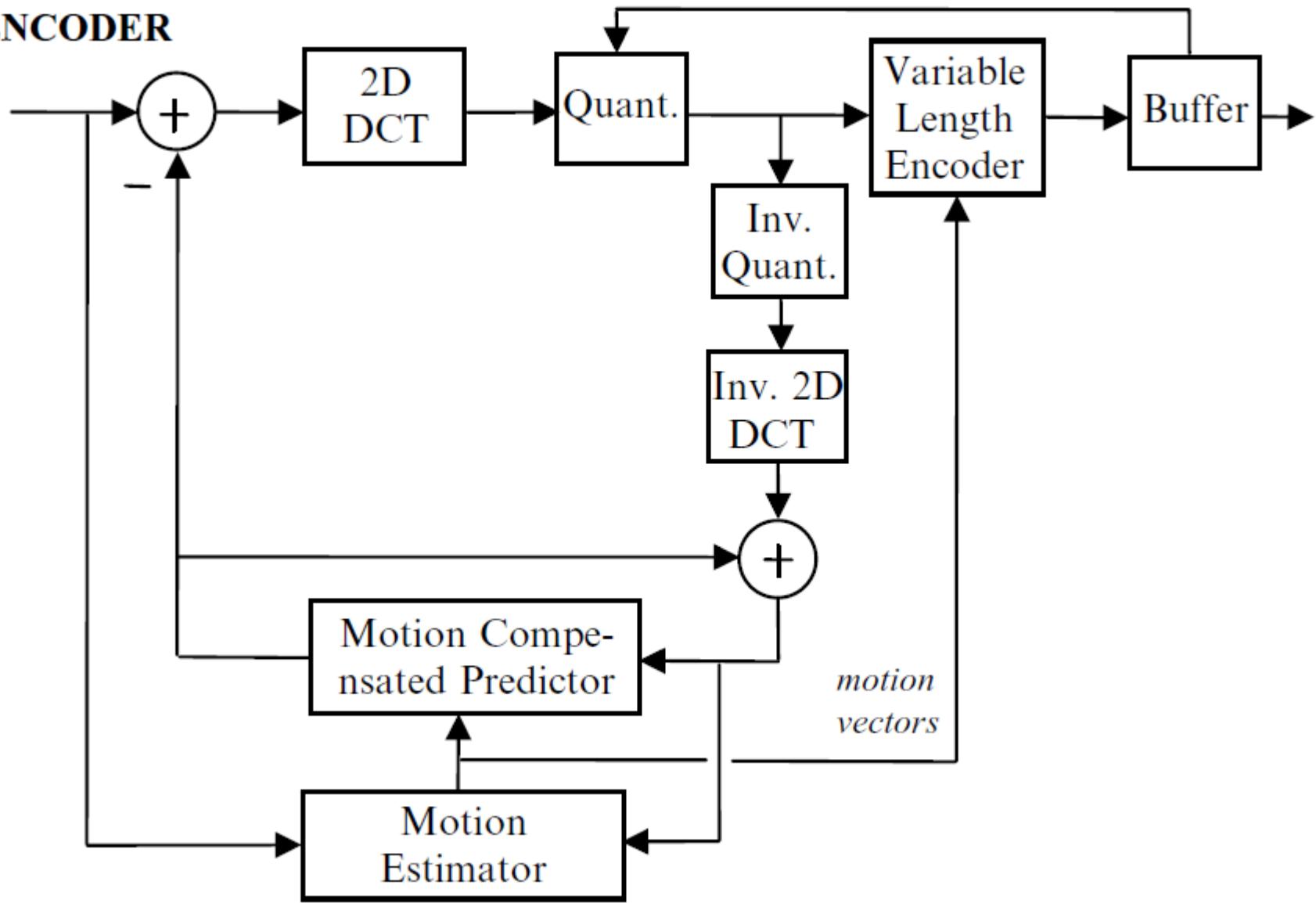
Blocos menores: melhor predição e mais vetores de movimento para codificar

MPEG

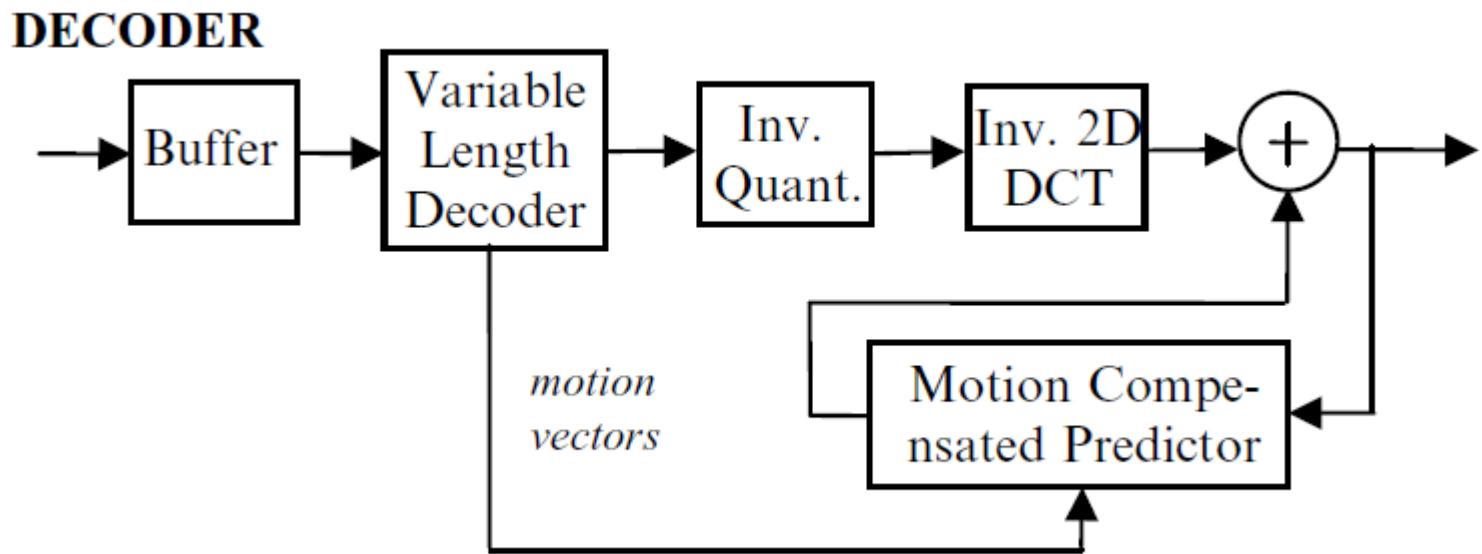


MPEG

ENCODER



MPEG



MP3

Camada 3 do MPEG1

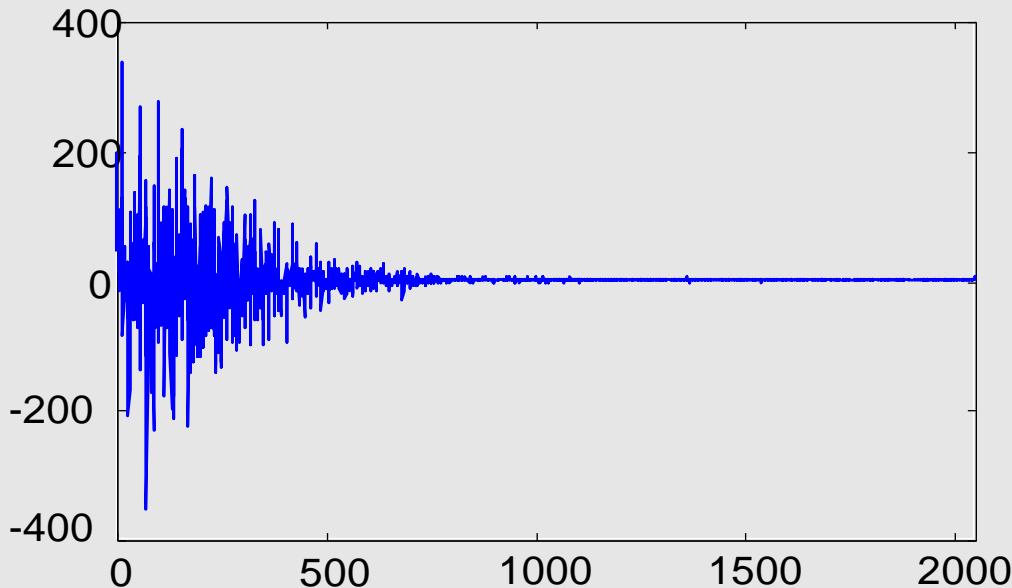
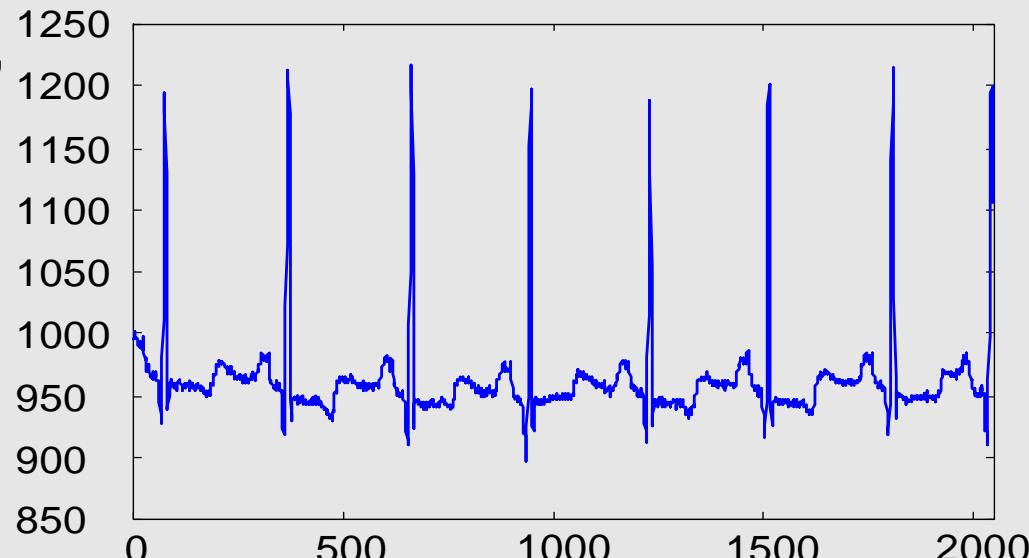
DCT em blocos, Quantização, Huffman

Máscaras perceptuais

DCT - Exemplo

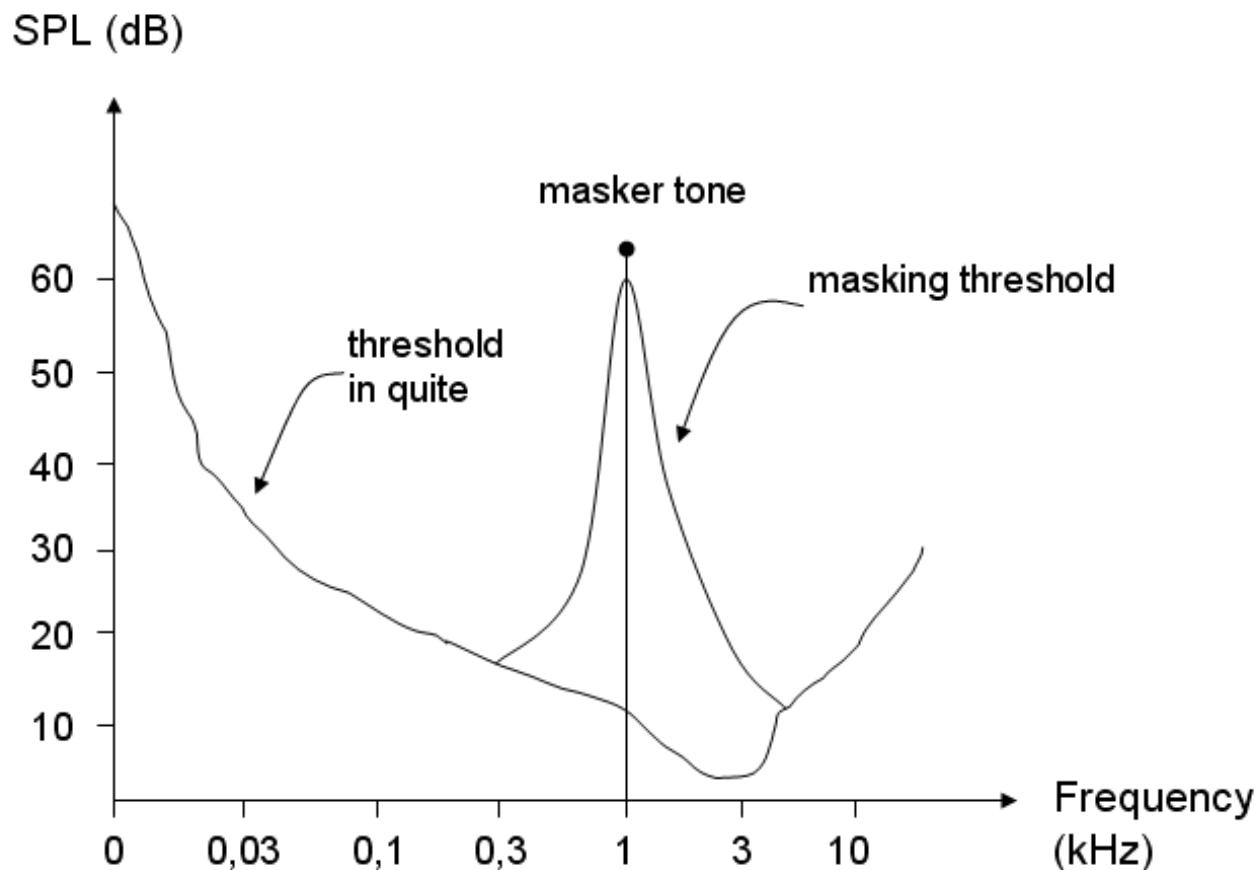
Sinal eletrocardiográfico,
2048 amostras

DCT do sinal
eletrocardiográfico
(sem coeficiente DC)



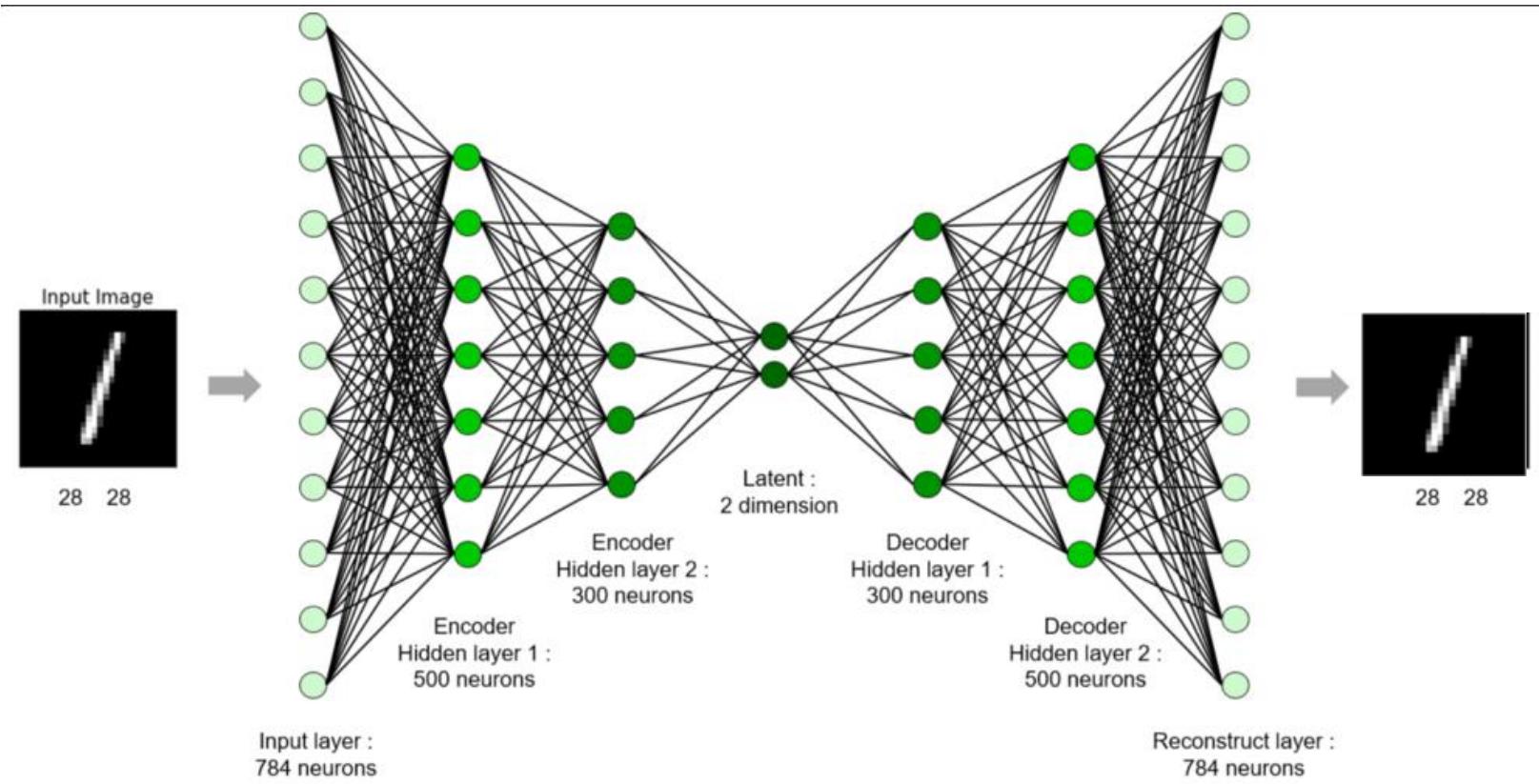
MP3

Máscaras perceptuais



Autoencoderes

Treinamento



Autoencoders

Teste

