

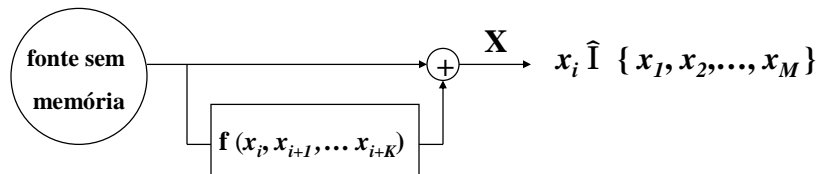


Sessão nº7

Fontes com Memória e Codificação Aritmética



Modelo de fonte discreta com memória



- x_i representa um símbolo da fonte
- $p(x_i)$ é a probabilidade dum símbolo tal que $\sum_{x_i \in X} p(x_i) = 1$
- $p(x_{i+1}|x_i)$ é a probabilidade dum símbolo dado o anterior
- $p(x_{i+2}|x_{i+1}, x_i)$ é a probabilidade dum símbolo dado os dois últimos
- símbolos dependentes (existe memória)



Modelo de Markov de 1ª ordem - exemplo



- Consideremos uma sequência de símbolos $\{x_1, x_2, \dots, x_M\}$ produzidos por uma fonte binária \mathbf{X} onde cada símbolo só depende do anterior, logo $p(x_{n+1} | x_n, x_{n-1}, \dots, x_1) = p(x_{n+1} | x_n)$
- Sendo $p(0|0) = 0,6$, $p(1|0) = 0,4$, $p(0|1) = 0,1$, $p(1|1) = 0,9$ e 0 o símbolo anterior inicial verifica-se que $p(0)=0,2$ e $p(1)=0,8$, logo $H(p_0) \approx 0,72$.

- Entropia da fonte dado que o símbolo anterior foi x_i

$$H(\mathbf{X}|x_i) = \sum_{x_{i+1} \in X} p(x_{i+1} | x_i) \log_2 1/p(x_{i+1} | x_i)$$

- Entropia da fonte dado que se conhece o símbolo anterior

$$H(\mathbf{X}) = \sum_{x_i \in X} p(x_i) \times p(x_{i+1} | x_i) \log_2 1/p(x_{i+1} | x_i) \approx 0,57$$



Entropia duma fonte discreta com memória



- Admitindo que cada símbolo só depende dos 2 últimos símbolos, respectivamente x_{i+1} e x_i (modelo de Markov de 2ª ordem) vem que

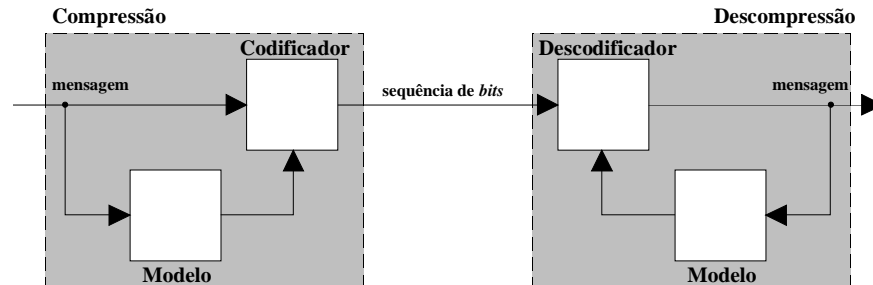
$$H(\mathbf{X}) = \sum_{x_i \in X} p(x_{i+1}, x_i) \times p(x_{i+2} | x_{i+1}, x_i) \log_2 1/p(x_{i+2} | x_{i+1}, x_i)$$

- Genericamente, admitindo que cada símbolo só depende dos m últimos símbolos (modelo de Markov de ordem m) vem que

$$H(\mathbf{X}) = \sum_{x_i \in X} p(x_{i+m-1}, \dots, x_{i+1}, x_i) \times p(x_{i+m} | x_{i+m-1}, \dots, x_{i+1}, x_i) \times \log_2 1/p(x_{i+m} | x_{i+m-1}, \dots, x_{i+1}, x_i)$$



Paradigma da compressão



- Representação **natural** (dimensão fixa) e representação **com compressão** (dimensão variável)
- Compromisso entre o **custo do espaço** e o **custo do tempo**



Ordem dos modelos



- Modelo livre de contexto ou de **ordem 0**
 - atribuição de probabilidades fixas para os símbolos da fonte, independentemente da sua posição no texto.
- Modelo de **ordem n**
 - estimativa das probabilidades por observação da interdependência dos símbolos, considerando os últimos n símbolos da fonte.
- Modelo de **ordem -1**
 - assumindo à priori uma distribuição de probabilidades, por exemplo equiprovável, independentemente do texto.



Memória necessária aos modelos



Ordem do modelo n	Quantidade de memória [byte]	Descrição
0	512	tabela de 256 entradas, com frequências de ocorrência.
1	131072	256 tabelas de 256 entradas cada, com frequências de ocorrência.
2	33554432	65536 tabelas de 256 entradas cada, com frequências de ocorrência.

- A fonte gera 256 símbolos diferentes, não independentes
- O valor da frequência de ocorrência é representado com 2 *byte*



Tipos de modelos



- **Modelo estático:** o mesmo modelo, determinado a partir duma amostra de texto conhecido por exemplo, é usado para todos os tipos de texto.
- **Modelo semi-adaptativo:** para cada mensagem utiliza-se o modelo obtido a partir da pré-análise do texto. Implica duas passagens sobre o texto.
- **Modelo adaptativo:** inicialmente tanto o codificador como o decodificador estão num estado conhecido (símbolos equiprováveis por ex.). Sempre que um símbolo é codificado ou decodificado, actualiza-se o modelo “incrementando” a sua probabilidade. O próximo símbolo já é codificado ou decodificado com o novo modelo e assim sucessivamente.



Comparação dos vários modelos



Modelo	Vantagens	Desvantagens
Estático	Simplicidade e velocidade de execução.	Quando há desajuste do modelo a codificação torna-se ineficiente.
Semi-adaptativo	Modelo adaptado aos dados.	Necessidade de fazer duas passagens sobre a mensagem. A transmissão do modelo é um custo extra que faz baixar a eficiência da compressão.
Adaptativo	Evita a transmissão do modelo; este fica adaptado aos dados só com uma passagem sobre a mensagem; a melhor eficiência de compressão.	No início do processo, tem baixa eficiência de compressão, sendo por isso ineficiente na compressão de textos com poucos símbolos.



Tipos de codificadores (quanto ao modelo)



- **Estatísticos**
 - a cada símbolo é atribuído um código de acordo com a sua probabilidade de ocorrência
 - **codificação de Huffman**
 - **codificação aritmética** (nos métodos mais eficiente é de uso comum)
- **Baseados em dicionário**
 - substituem-se grupos de símbolos consecutivos, por uma referência para o dicionário (código)
 - **codificação de Lempel-Ziv** (nos métodos mais eficiente é de uso comum)



Tipos de codificadores (quanto à fonte)

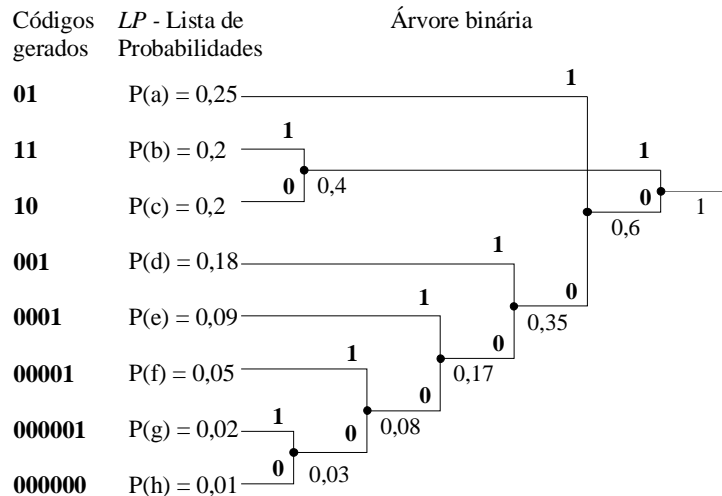


- **Codificação com modelo de fonte**
 - codificação de Huffman (codificação óptima para fontes estacionárias)

- **Codificação Universal de fonte**
 - codificação aritmética
 - codificação de Lempel-Ziv



Codificação de Huffman



Codificação Aritmética - algoritmo



- Atribuir ao intervalo de probabilidades corrente $[L, H[$ o valor $[0, 1[$
- Para cada símbolo a codificar:
 - subdividir o intervalo corrente em subintervalos, um para cada símbolo possível na mensagem. A largura do subintervalo associado a um símbolo é proporcional à probabilidade estimada para que esse símbolo seja o próximo símbolo a codificar, de acordo com o modelo da fonte
 - seleccionar o subintervalo correspondente ao próximo símbolo e considerar esse o novo intervalo corrente
- Codificar o intervalo corrente final com o número de bits suficientes para o distinguir de todos os outros intervalos possíveis

(Langdon 1984)

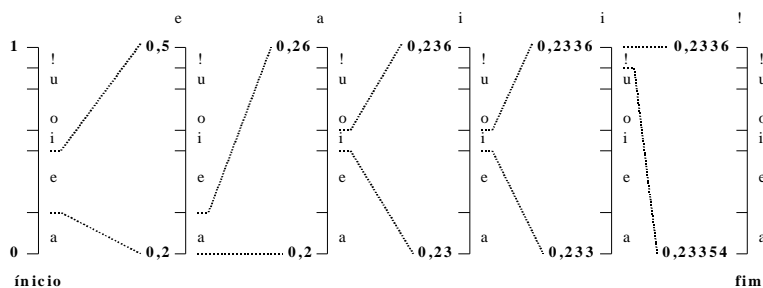


Codificação Aritmética - exemplo



Símbolo	Probabilidade	Intervalo
a	0,2	$[0, .0,2[$
e	0,3	$[0,2, .0,5[$
i	0,1	$[0,5, .0,6[$
o	0,2	$[0,6, .0,8[$
u	0,1	$[0,8, .0,9[$
!	0,1	$[0,9, .1,0[$

- Modelo utilizado na codificação da sequência “eaii!”
- Representação gráfica do processo de codificação



Codificação Aritmética - implementação



- **Codificação**

```
low = 0 ; high = 0 ; range = 1 ;
```

```
loop
```

```
high = low + range × Higher( simb ) ;
```

```
low = low + range × Lower( simb ) ;
```

```
range = high - low ;
```

```
outputCode = low ;
```

- **Descodificação**

```
n = outputCode ;
```

```
loop
```

```
simb = GetSimb(n) ;
```

```
n = ( n - Lower( simb ) ) / Range( simb ) ;
```



Transformada de Burrows-Wheeler (BWT)



- **Algoritmo** ao nível do bloco (símbolos rearranjados apenas)
- Transformada **reversível**
- Texto original - 1 *Kbyte* (excerto)

```
the problems of philosophy bertrand russell the problems of philosophy be  
rtrand russell oxford university press london oxford new york first publish  
ed in the home university library first issued as an oxford university pr  
ess paperback this reprint printed in the united states of ameri
```

- BWT do texto (excerto)

```
rbvhrvhmrlhlmhcyrdfrbrldcnuthtr rbh ssecllt rmmrrhytphmf hvvvvdbbiiitr  
lnt hoo of foi xxxe annnn iii ttnt ttttttsttttpttttt tgg gpp cllnrtrndrrrtwh  
h aaa rdshrr tsteff hhhhh l rnwwdssnnn crllillmm obebbp bbbeeian iieoo  
aoo eattooebiiiiioioaiioaauaaaaeuoo iiiieoauuuuw a eiueiuetdlrrr
```



Entropia estimada depois da BWT



Calgary Corpus	$\hat{E}(X)$	$\hat{E}_{BWT}(X)$	$\hat{E}(X,Y)/2$	$\hat{E}_{BWT}(X,Y)/2$
BIB	5,20	5,20	4,28	3,74
BOOK1	4,53	4,53	4,06	3,83
BOOK2	4,79	4,79	4,27	3,75
GEO	5,65	5,65	4,96	4,75
NEWS	5,19	5,19	4,64	4,12
OBJ1	5,95	5,95	4,71	4,57
OBJ2	6,26	6,26	5,07	4,44
PAPER1	4,98	4,99	4,31	3,86
PAPER2	4,60	4,60	4,06	3,69
PAPER3	4,67	4,67	4,11	3,81
PAPER4	4,70	4,71	4,09	3,88
PAPER5	4,94	4,94	4,23	3,99
PAPER6	5,01	5,01	4,31	3,87
PIC	1,21	1,21	1,02	1,00
PROGC	5,20	5,20	4,40	3,94
PROGL	4,77	4,77	3,99	3,36
PROGP	4,87	4,87	4,03	3,36
TRANS/BWT	5,53	5,53	4,44	3,57
valor médio	4,89	4,89	4,17	3,75



Entropia estimada depois da BWT+MTF+RLE



Calgary Corpus	$\hat{E}(X)$	$\hat{E}_{BMR}(X)$	$\hat{E}(X,Y)/2$	$\hat{E}_{BMR}(X,Y)/2$
BIB	5,20	3,60	4,28	3,55
BOOK1	4,53	3,39	4,06	3,35
PAPER2	4,60	3,45	4,06	3,40



Eficácia de compressão depois da BWT



Tipo de Compressor	Calg. Corpus [bits/byte]	BWT(Calg. Corpus) [bits/byte]
Codificador de Huffamn	4,99	5,00
Codificador Aritmético	4,95	4,96
Codificador Aritmético (bigramas)	4,16	3,30

