Codificação de Fonte
(Códigos de Shannon e Shannon-Fano)

Teoria da Informação - AULA 11

Profª. Verusca Severo

Universidade de Pernambuco
Escola Politécnica de Pernambuco

04 de agosto de 2021

Introdução

- Discutimos nas aulas anteriores a importância da eficiência de um código.
- Vimos que essa eficiência está associada à economia que se usa na escolha dos comprimentos das palavras código.
- Como, em geral, os símbolos de uma fonte não são equiprováveis, é razoável pensar em códigos de comprimento variável, permitindo com que símbolos que ocorrem mais vezes (com maior probabilidade de ocorrência) recebam palavras código de menor tamanho.

Introdução

- Nesse sentido, no processo de codificação, queremos transformar as mensagens de uma fonte em um conjunto de símbolos de modo que:
    - seja ocupado menos espaço (reduzindo os requisitos de memória para armazenamento);
    - apresente uma menor taxa de codificação (reduzindo os requisitos de largura de banda para transmissão).
- Queremos fazer isso "SEM PERDAS", ou seja, a decodificação deve fornecer exatamente as mesmas mensagens originais.

Introdução

- Logo, a questão está em descobrir como codificar eficientemente os símbolos da fonte, tendo cada um uma certa probabilidade de ocorrência.

- No caso dos símbolos da fonte não serem equiprováveis, o código ótimo tem em conta as diferentes probabilidades de ocorrência dos símbolos, usando palavras código de comprimento variável.

Teorema da Codificação de Shannon

- Na aula anterior, provamos que qualquer conjunto de palavras-código que satisfaça a condição de código prefixo deve satisfazer a desigualdade de Kraft
  - a desigualdade é uma condição suficiente para a existência de um código prefixo com um conjunto de palavras-código com especificado comprimento.

- Agora consideramos o problema de encontrar o código prefixo com o comprimento mínimo esperado.

- **Teorema da codificação:** um código D-ário de prefixo ótimo de uma variável aleatória U deve satisfazer a relação

$$H(U) \leq L < H(U) + 1$$

Prova (Teorema da codificação):

- O problema de encontrar um código prefixo com o comprimento mínimo esperado é equivalente a encontrar o conjunto de comprimentos $l_1, l_2, \ldots, l_K$, satisfazendo a desigualdade de Kraft e cujo comprimento esperado $L = \sum p_i l_i$ é menor do que o comprimento esperado de qualquer outro código prefixo.

- Este é um problema de otimização padrão:
  - Minimize

$$L = \sum p_i l_i$$

sobre todos os inteiros $l_1, l_2, \ldots, l_K$ satisfazendo

$$\sum D^{-l_i} \leq 1$$

Prova (Teorema da codificação):

- Podemos escrever essa minimização usando multiplicadores de Lagrange:
$$J = \sum p_i l_i + \lambda(\sum D^{-l_i})$$
- Resolvendo esse problema de minimização, chegamos na seguinte solução
$$p_i = D^{-l_i},$$
produzindo comprimentos de código ideais
$$l_i = -\log_D(p_i) = -\log_D(D^{-l_i})$$
- Mas como l_i deve ser inteiro, nem sempre seremos capazes de definir os comprimentos da palavra código como descrito acima. Em vez disso, devemos escolher um conjunto de comprimentos de palavra código l_i "próximos" ao conjunto ideal.

Prova (Teorema da codificação):

- Daí temos que:

$$
\begin{aligned}
L &\geq H(U) \qquad (1)\\
L - H(U) &\geq 0\\
L - H(U) &= \sum p_i l_i - \sum p_i \log_D \left(\frac{1}{p_i}\right)\\
L - H(U) &= \sum p_i l_i + \sum p_i \log_D p_i\\
L - H(U) &= \sum p_i \times (-\log_D(D^{-l_i})) + \sum p_i \log_D p_i\\
L - H(U) &= -\sum p_i \log_D(D^{-l_i}) + \sum p_i \log_D p_i
\end{aligned}
$$

- Lembre-se, l_i deve ser inteiro. Para isso devemos escolher um conjunto de comprimentos de palavra código l_i “próximos” ao conjunto ideal (o inteiro acima mais próximo).

Prova (Teorema da codificação):

- Portanto, a expressão em destaque será sempre maior igual a parcela da entropia.

$$L - H(U) = -\sum p_i \log_D(D^{-l_i}) + \sum p_i \log_D p_i \tag{2}$$
$$L - H(U) \geq 0$$

- Haverá a igualdade quando $D^{-l_i} = p_i$

- Há um "overhead" de no máximo 1 *bit* no lado direito da expressão em $H(U) \leq L < H(U) + 1$ devido ao fato de que l_i nem sempre é um número inteiro.

Teorema da Codificação de Shannon

- Pelo Teorema da codificação, temos que uma codificação é dita ótima quando:

$$L = H(U)$$

- A eficiência da codificação mede-se calculando a razão

$$\eta = \frac{H(U)}{L}$$

Código de Shannon

- Como aumentar a eficiência de um código?

Teorema da Codificação de Shannon

- Para aumentar a eficiência é preciso diminuir o comprimento médio L, o que se consegue com os códigos de comprimento variável.
  - Aos símbolos que ocorrem com mais frequência devem corresponder palavras código mais curtas que as que correspondem aos símbolos mais raros.

- **Exemplo 1:** Seja uma fonte discreta sem memória que emite símbolos do alfabeto $S = \{s_1, s_2, s_3, s_4\}$ com distribuição de probabilidades $P(S = s_1) = \frac{1}{2}$, $P(S = s_2) = \frac{1}{4}$, $P(S = s_3) = \frac{1}{8}$ e $P(S = s_4) = \frac{1}{8}$. Construa um código de comprimento fixo para esta fonte, calcule a sua eficiência e verifique se ele satisfaz o teorema da codificação.

Solução (Exemplo 1):

$$\#bits = \log_2 4 = 2bits \text{ por símbolo}$$

- Um possível código é:

$$\begin{array}{ll} s_1 \Rightarrow 00 & s_2 \Rightarrow 01 \\ s_3 \Rightarrow 10 & s_4 \Rightarrow 11 \end{array}$$

- **Eficiência do código:**

$$H(S) = \frac{1}{2}\log_2\frac{1}{\frac{1}{2}} + \frac{1}{4}\log_2\frac{1}{\frac{1}{4}} + \frac{1}{8}\log_2\frac{1}{\frac{1}{8}} + \frac{1}{8}\log_2\frac{1}{\frac{1}{8}} = 1,75$$

$$L = 2\times\frac{1}{2} + 2\times\frac{1}{4} + 2\times\frac{1}{8} + 2\times\frac{1}{8} = 2$$

$$\eta = \frac{H(S)}{L} = \frac{1,75}{2} = 0,875\times 100\% = 87,5\%$$

Solução (Exemplo 1):

- Verificando se o código satisfaz o teorema da codificação:

$$H(U) \leq L < H(U) + 1$$

$$1,75 \leq 2 < 2,75$$

Código de Shannon

- Na busca de um código prefixo ótimo (no sentido de alcançar um código em que $L = H(U)$, Shannon propôs um código em que o comprimento de cada palavra código deve ser determinada pela expressão

$$l_i = \left\lceil \log_D \frac{1}{p_i} \right\rceil,$$

em que p_i corresponde a probabilidade de ocorrência do símbolo.

  - **OBS.:** na expressão acima, a função "teto" é utilizada para fornecer o comprimento da palavra inteira, uma vez que $\log_D \frac{1}{p_i}$ pode não ser igual a um inteiro.

- **Exemplo 2:** Construa um código do tipo Shannon para a fonte apresentada no **Exemplo 1**, calcule a eficiência do código obtido e verifique se ele satisfaz o teorema da codificação.

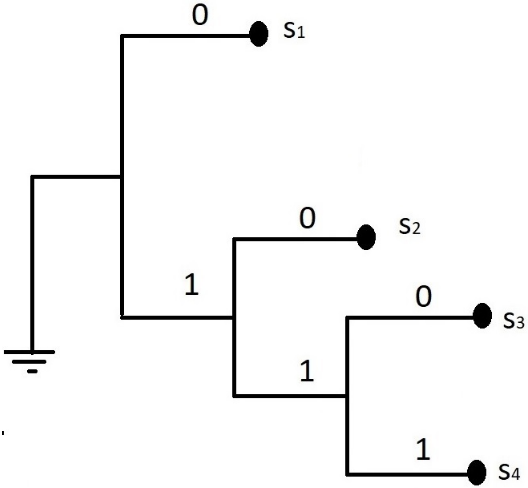
Solução (Exemplo 2):

- Precisamos determinar o comprimento de cada palavra código:
  - Para s_1: $l_1 = \left\lceil \log_2 \frac{1}{\frac{1}{2}} \right\rceil = \lceil \log_2 2 \rceil = 1$
  - Para s_2: $l_2 = \left\lceil \log_2 \frac{1}{\frac{1}{4}} \right\rceil = \lceil \log_2 4 \rceil = 2$
  - Para s_3: $l_3 = \left\lceil \log_2 \frac{1}{\frac{1}{8}} \right\rceil = \lceil \log_2 8 \rceil = 3$
  - Para s_4: $l_4 = \left\lceil \log_2 \frac{1}{\frac{1}{8}} \right\rceil = \lceil \log_2 8 \rceil = 3$

- Um possível código é:

$$s_1 \Rightarrow 0$$
$$s_2 \Rightarrow 10$$
$$s_3 \Rightarrow 110$$
$$s_4 \Rightarrow 111$$

Solução (Exemplo 2):



Solução (Exemplo 2):

- **Eficiência do código:**

$$H(S) = \frac{1}{2}\log_2\frac{1}{\frac{1}{2}} + \frac{1}{4}\log_2\frac{1}{\frac{1}{4}} + \frac{1}{8}\log_2\frac{1}{\frac{1}{8}} + \frac{1}{8}\log_2\frac{1}{\frac{1}{8}} = 1,75$$

$$L = 1 \times \frac{1}{2} + 2 \times \frac{1}{4} + 3 \times \frac{1}{8} + 3 \times \frac{1}{8} = 1,75$$

- Logo:

$$\eta = \frac{H(S)}{L} = \frac{1,75}{1,75} = 1 \times 100\% = 100\%$$

Solução (Exemplo 2):

- Verificando se o código satisfaz o teorema da codificação:

$$H(U) \leq L < H(U) + 1$$

$$1,75 \leq 1,75 < 2,75$$

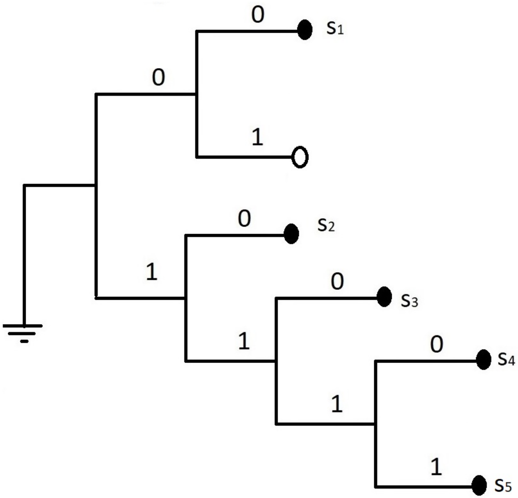
- **Exemplo 3:** Seja uma fonte discreta sem memória que emite símbolos do alfabeto $S = \{s_1, s_2, s_3, s_4, s_5\}$ com distribuição de probabilidades $P(S = s_1) = 0,3$, $P(S = s_2) = 0,3$, $P(S = s_3) = 0,2$, $P(S = s_4) = 0,1$ e $P(S = s_5) = 0,1$. Construa um código do tipo Shannon para a esta fonte, calcule a eficiência do código obtido e verifique se ele satisfaz o teorema da codificação.

Solução (Exemplo 3):

- Precisamos determinar o comprimento de cada palavra código:
  - Para s_1: $l_1 = \left\lceil \log_2 \frac{1}{0{,}3} \right\rceil = 2$
  - Para s_2: $l_2 = \left\lceil \log_2 \frac{1}{0{,}3} \right\rceil = 2$
  - Para s_3: $l_3 = \left\lceil \log_2 \frac{1}{0{,}2} \right\rceil = 3$
  - Para s_4: $l_4 = \left\lceil \log_2 \frac{1}{0{,}1} \right\rceil = 4$
  - Para s_5: $l_5 = \left\lceil \log_2 \frac{1}{0{,}1} \right\rceil = 4$
- Um possível código é:

$$
\begin{aligned}
s_1 &\Rightarrow 00 \\
s_2 &\Rightarrow 10 \\
s_3 &\Rightarrow 110 \\
s_4 &\Rightarrow 1110 \\
s_5 &\Rightarrow 1111
\end{aligned}
$$

Solução (Exemplo 3):



Solução (Exemplo 3):

- **Eficiência do código:**

$$H(S) = 2 \times (0,3 \log_2 \frac{1}{0,3}) + 0,2 \log_2 \frac{1}{0,2} + 2 \times (0,1 \log_2 \frac{1}{0,1}) = 2,17$$

$$L = 2 \times 0,3 + 2 \times 0,3 + 3 \times 0,2 + 4 \times 0,1 + 4 \times 0,1 = 2,8$$

- Logo:

$$\eta = \frac{H(S)}{L} = \frac{2,17}{2,8} = 0,775 \times 100\% = 77,5\%$$

Código de Shannon

Solução (Exemplo 3):

- Verificando se o código satisfaz o teorema da codificação:

$$H(U) \leq L < H(U) + 1$$

$$2,17 \leq 2,8 < 3,17$$

Código de Shannon

- **Exemplo 4:** Seja uma fonte discreta sem memória que emite símbolos do alfabeto $S = \{s_1, s_2\}$ com distribuição de probabilidades $P(S = s_1) = \frac{3}{4}$ e $P(S = s_2) = \frac{1}{4}$.
  a. Construa um código de comprimento fixo para esta fonte e calcule a eficiência do código obtido.
  b. Construa um código de Shannon para esta fonte e calcule a eficiência do código obtido.

Solução (Exemplo 4a):

$$\#bits = \log_2 2 = 1bit \text{ por símbolo}$$

- Um possível código é:

$$s_1 \Rightarrow 0$$
$$s_2 \Rightarrow 1$$

- **Eficiência do código:**

$$H(S) = \frac{3}{4}\log_2 \frac{1}{\frac{3}{4}} + \frac{1}{4}\log_2 \frac{1}{\frac{1}{4}} = 0,811$$

$$L = 1 \times \frac{3}{4} + 1 \times \frac{1}{4} = 1$$

$$\eta = \frac{H(S)}{L} = \frac{0,811}{1} = 0,811 \times 100\% = 81,1\%$$

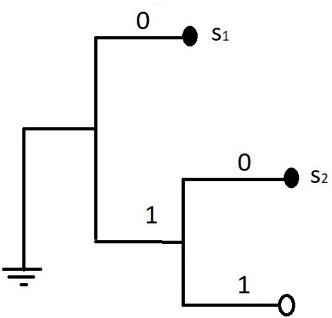
Solução (Exemplo 4b):

- Precisamos determinar o comprimento de cada palavra código:
  - Para s_1: $l_1 = \left\lceil \log_2 \frac{1}{\frac{3}{4}} \right\rceil = 1$
  - Para s_2: $l_2 = \left\lceil \log_2 \frac{1}{\frac{1}{4}} \right\rceil = 2$

- Um possível código é:

$$s_1 \Rightarrow 0$$
$$s_2 \Rightarrow 10$$

Solução (Exemplo 4b):



Solução (Exemplo 4b):

- **Eficiência do código:**

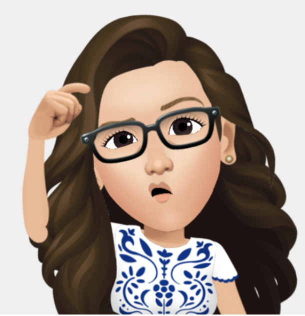
$$H(S) = \frac{3}{4}\log_2\frac{1}{\frac{3}{4}} + \frac{1}{4}\log_2\frac{1}{\frac{1}{4}} = 0,811$$

$$L = 1 \times \frac{3}{4} + 2 \times \frac{1}{4} = 1,25$$

$$\eta = \frac{H(S)}{L} = \frac{0,811}{1,25} = 0,6488 \times 100\% = 64,88\%$$

Código de Shannon

- Existe algum código para a fonte do **Exemplo 4**, com comprimento médio das palavras código menor do que 1 ?



- Vamos analisar a expressão que define o teorema da codificação:

$$H(U) \leq L < H(U) + 1$$

- Vimos, na aula 05, que a extensão de fonte sem memória apresenta entropia de

$$H(U^n) = n \times H(U)$$

Código de Shannon

- Utilizando fontes extendidas, temos que o teorema da codificação resultará em:

$$H(U^n) \leq L_n < H(U^n) + 1$$

$$n \times H(U) \leq L_n < n \times H(U) + 1$$

- Para eliminar n, devemos realizar uma divisão por n, Logo:

$$\frac{n \times H(U)}{n} \leq \frac{L_n}{n} < \frac{n \times H(U) + 1}{n}$$

$$H(U) \leq \frac{L_n}{n} < H(U) + \frac{1}{n},$$

em que $H(U^n)$ denota a entropia da extensão de ordem n da fonte U e $\frac{L_n}{n}$ denota o comprimento médio das palavras código por bloco de n dígitos da fonte.

- Logo, respondendo ao questionamento: Existe algum código para a fonte do **Exemplo 4**, com comprimento médio das palavras código menor do que 1 ?:
    - Pelo teorema da codificação temos que $H(U) \approx \frac{L_n}{n}$.
    - Para o exexmplo 4, temos $H(U) = 0,811$ e portanto é possível encontrar um código univocamente decodificável com comprimento médio das palavras código menor que 1.

- Logo, respondendo ao questionamento: Existe algum código para a fonte do **Exemplo 4**, com comprimento médio das palavras código menor do que 1 ?:
  - **EXEMPLO 4c:** A extensão de ordem $n = 2$ da fonte dada fornece o seguinte alfabeto:

U^2	**Probabilidade**
$s_1 s_1$	$\frac{3}{4} \times \frac{3}{4} = \frac{9}{16}$
$s_1 s_2$	$\frac{3}{4} \times \frac{1}{4} = \frac{3}{16}$
$s_2 s_1$	$\frac{1}{4} \times \frac{3}{4} = \frac{3}{16}$
$s_2 s_2$	$\frac{1}{4} \times \frac{1}{4} = \frac{1}{16}$

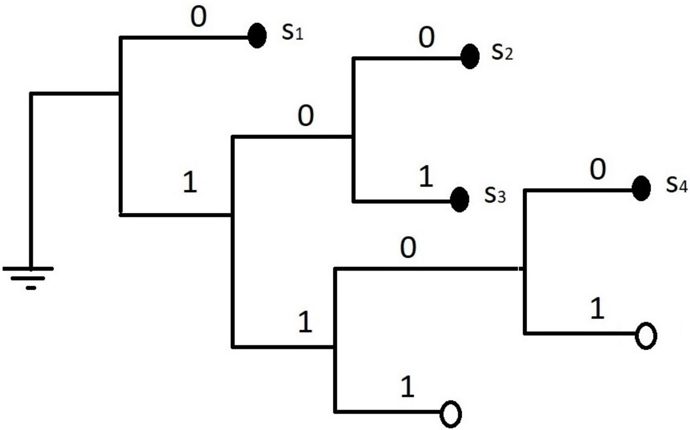
Código de Shannon

- Precisamos determinar o comprimento de cada palavra código para o código de Shannon:
  - Para $s_1 s_1$: $l_1 = \left\lceil \log_2 \frac{1}{\frac{9}{16}} \right\rceil = 1$
  - Para $s_1 s_2$: $l_2 = \left\lceil \log_2 \frac{1}{\frac{3}{16}} \right\rceil = 3$
  - Para $s_2 s_1$: $l_3 = \left\lceil \log_2 \frac{1}{\frac{3}{16}} \right\rceil = 3$
  - Para $s_2 s_2$: $l_4 = \left\lceil \log_2 \frac{1}{\frac{1}{16}} \right\rceil = 4$

- Um possível código é:

$$
\begin{aligned}
s_1 &\Rightarrow 0 \\
s_2 &\Rightarrow 100 \\
s_3 &\Rightarrow 101 \\
s_4 &\Rightarrow 1100
\end{aligned}
$$

Código de Shannon



- Logo, a extensão de ordem $n = 2$ da fote U permite a construção de um código univocamente decodificável para o qual o comprimento médio das palavras código por bloco de n dígitos da fonte é $\frac{L_2}{2} = \frac{31/16}{2} = \frac{1{,}9375}{2} = 0,96875$

- **Eficiência do código:**

$$H(U^2) = 2 \times H(U) = 2 \times 0,811 = 1,622$$

$$L_2 = 1 \times \frac{9}{16} + 3 \times \frac{3}{16} + 3 \times \frac{3}{16} + 4 \times \frac{1}{16} = 1,9375$$

$$\eta = \frac{n \times H(U)}{L_n} = \frac{2 \times H(U)}{L_2} = \frac{1,622}{1,9375} = 0,8372 \times 100\% = 83,72\%$$

Código de Shannon-Fano

- O método Shannon-Fano foi apresentado por Shannon e Fano em 1949.
- O objetivo deste método é associar códigos menores a símbolos mais prováveis e códigos maiores aos menos prováveis.
- A codificação parte da construção de uma árvore ponderada considerando a probabilidade de ocorrência de cada símbolo.

Código de Shannon-Fano

- O código é feito da seguinte forma:

1. Monte uma lista com os símbolos em ordem decrescente de probabilidade. Esta lista é associada à raiz da árvore;
2. Divida a lista em duas sublistas de tal maneira que as somas das probabilidades de cada uma delas seja mínima, isto é, a soma de uma delas deve ser aproximadamente igual à soma da outra;
3. As sublistas serão então os filhos do nó anterior. Atribua o *bit* 0 como primeiro dígito das palavras código associadas as mensagens na primeira sublista e atribua o *bit* 1 a segunda sublista.
4. Os passos 2 e 3 acima são repetidos até que as sublistas sejam unitárias.

- **Exemplo 5:** Seja uma fonte discreta sem memória que emite símbolos do alfabeto $S = \{s_1, s_2, s_3, s_4, s_5\}$ com distribuição de probabilidades $P(S = s_1) = 0,4$, $P(S = s_2) = 0,2$, $P(S = s_3) = 0,2$, $P(S = s_4) = 0,1$ e $P(S = s_5) = 0,1$.
  a. Construa um código de Shannon-Fano para esta fonte e calcule a eficiência do código obtido.

Solução (Exemplo 5):

VER MATERIAL EM ANEXO!

- **Exemplo 6:** Construa um código de Shannon-Fano para a fonte do **Exemplo 3** e calcule a eficiência do código obtido.

Solução (Exemplo 6):

VER MATERIAL EM ANEXO!

Código de Shannon

- **Exemplo 7:** Refaça o **Exemplo 4c** com o código de Shannon-Fano

Solução (Exemplo 7):

VER MATERIAL EM ANEXO!

- **Exercício:** Refaça o **Exemplo 7** mas considerando a fonte U^3 (extensão de ordem $n = 3$ da fonte U do exemplo 4)

Solução (Exercício):

VER MATERIAL EM ANEXO!