




SCC261 - Multimídia

Aula 3 – Introdução à Teoria da Informação e Técnicas de Compressão


Prof.: Dr. Rudinei Goularte
 (rudinei@icmc.usp.br)
 Instituto de Ciências Matemáticas e de Computação - ICMC
 Sala 4-234




Sumário

- 1. Introdução a Técnicas de Compressão
- 2. Compressão Sem Perdas

2




1 – Introdução a Técnicas de Compressão



1.1 Por quê Comprimir?

- Preencher o "gap" demanda x capacidade
 - Usuários têm demandado aplicações com mídias cada vez mais sofisticadas.
 - Meios de transmissão e armazenamento são limitados.
 - Livro de 800 páginas. Cada página com 40 linhas. Cada linha com 80 caracteres. $\rightarrow 800 * 40 * 80 = (1 \text{ byte por caractere}) 2,44 \text{ MB}$.
 - Vídeo digital com "qualidade de TV" (aproximadamente):
 - 1 segundo = 216Mbits.
 - 2 horas = 194GB = 42 DVDs ou 304 CD-ROMs!
 - "Compressão vai se tornar redundante em breve, conforme as capacidades de armazenamento e transmissão aumentem."
 - Esta frase tem sido repetida nos últimos 25 anos.


4



1.2 Codificador e Decodificador

- Compressão x Descompressão.
 - (A)simetria
- Implementação do algoritmo de compressão:
 - Em software ou em hardware.

5



1.3 Tipos de Algoritmos de Compressão

- Compressão sem perdas: Lossless.
 - Reversível.
 - Exemplos.
- Compressão com perdas: Lossy.
 - Não reversível.
 - Taxas de compressão x qualidade.
 - Exemplos.

6

2. Compressão Sem Perdas

Qual a quantidade mínima de bits necessária para representar uma informação sem que ocorram perdas?

2.1 Teoria da Informação

- Ferramenta matemática para determinar a quantidade mínima de dados para representar informação.
 - Premissa: a geração da informação pode ser modelada como um processo probabilístico.
 - Incerteza.

8

2.1 Teoria da Informação

- Incerteza

Se $P(E) = 1 \Rightarrow I(E) = 0$

- $I(E) = \log \frac{1}{P(E)} = -\log P(E)$ unidades de informação

9

2.1 Teoria da Informação

- Teorema de Shannon
 - Transferência de informação em canais de comunicação.
 - Fórmula de Shannon:
$$H = -\sum_{i=1}^n P_i \log_2 P_i$$
 - n = número de diferentes símbolos; P_i = probabilidade de ocorrência do símbolo i .

10

2.1 Teoria da Informação

- Eficiência de um esquema de compressão:
 - Entropia da fonte x Número médio de bits por código do esquema.
 - Número médio de bits por código:
$$NMB = \sum_{i=1} N_i P_i$$

11

2.1 Teoria da Informação

- Exemplo.
 - Novo método de compressão. Alfabeto: M, F, Y, N, 0 e 1. Frequência: 0.25, 0.25, 0.125, 0.125, 0.125 e 0.125. Códigos: M = 10, F = 11, Y = 010, N = 011, 0 = 000, 1 = 001.
 - A) Qual a entropia da fonte?
 - B) Qual o número médio de bits por código?

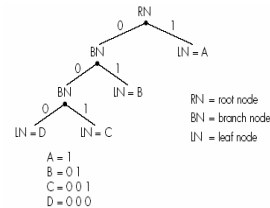
12

2.2 Codificação de Huffman

- Conjunto de dados e frequência relativa dos símbolos.
- Árvore binária não balanceada.
 - Os símbolos estão nas folhas.
 - Árvore de Huffman.
- Exemplos com caracteres.

13

2.2 Codificação de Huffman



- Seja a string AAAABBCD:
 - Códigos de Huffman requerem, nesse caso, 14 bits.
 - Versus 64 bits do código ASCII.

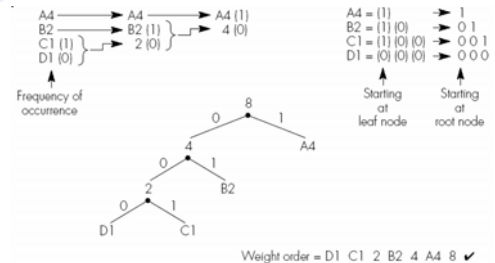
14

2.2 Codificação de Huffman

- Para construir uma árvore de Huffman é necessário obter a frequência dos símbolos:
 - Os símbolos com maior frequência devem ter os menores códigos.
 - Monta-se uma lista ordenada pela frequência:
 - A4
 - B2
 - C1
 - D1

15

2.2 Codificação de Huffman



16

2.2 Codificação de Huffman

- Árvore ótima (de Huffman).
- Árvore de Huffman tem a propriedade do prefixo.
 - Nenhum código é prefixo de outro código.

17

2.2 Codificação de Huffman

- Codificação:
 - AAAABBCD será 11110101001000
- Decodificação:
 - Árvore como índice.
 - Percurso da raiz para as folhas.

18

2.2 Codificação de Huffman

- Observações:
 - Ambos, codificador e decodificador devem conhecer a tabela de códigos.
 - Se a tabela é enviada/codificada junto com os dados, ocorre overhead.
 - O decodificador pode conhecer a tabela com antecedência.
 - Análise estatística do uso dos caracteres em uma determinada língua.
 - Esse método não é exato.
 - Alguns textos não vão atingir o máximo de compressão que poderiam.

19

Exercício

- Seja uma tabela de frequências relativas como segue:
 - A e B = 0,25; C e D = 0,14; E, F, G e H = 0,055.
- Derive um conjunto de códigos usando o método de Huffman.
- Derive o número médio de bits por carácter de seu código e compare com:
 - A entropia da fonte.
 - Um código binário de tamanho fixo.
 - Códigos ASCII de 7 bits.

20

2.3 Codificação por Diferença

- Codificação por diferença
 - Quando usar?
 - O quê codificar?
 - Perdas?

21

2.3 Codificação por Diferença

- Exemplo:
 - 12, 13, 11, 11, 10, ...

22

2.3 Codificação por Diferença

Number of bits needed (SSS)	Huffman codeword
0	010
1	011
2	100
3	00
4	101
5	110
6	1110
7	11110
11	11111110

23

2.4 Codificação por Carreira

- Também chamada *Run-Length Encoding* (RLE)
 - (skip, value)
 - 6, 7, 3, 3, 3, 2, 2, 2, 0, 0, 0, ...0.

24

2.4 Codificação por Carreira

- Exemplo:
 - 6, 7, 0, 0, 0, 3, -1, 0, ..., 0.

25

2.5 Codificação Aritmética

- Método de Huffman atinge o valor da Entropia apenas em algumas situações.
 - Depende da probabilidade de aparecimento dos caracteres no texto.
- Codificação Aritmética sempre atinge o valor da Entropia.
 - Mais complexa que Huffman.
 - Iremos estudar apenas o modo básico.

26

2.5 Codificação Aritmética

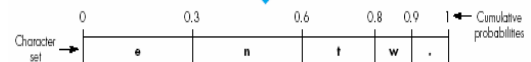
- String a ser codificada: went
- Probabilidades:
 - e = 0,3; n = 0,3; t = 0,2; w = 0,1; . = 0,1
 - . = terminador de string
- Conjunto de caracteres deve ser dividido no intervalo de 0 a 1, respeitando-se a proporção das probabilidades.

27

2.5 Codificação Aritmética

Example character set and their probabilities:

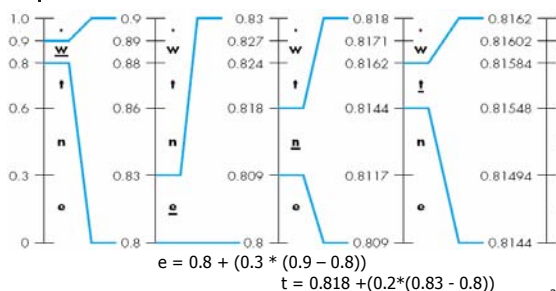
e = 0.3, n = 0.3, t = 0.2, w = 0.1, . = 0.1



- Cada subintervalo, na ordem da mensagem, é subdividido respeitando-se as proporções.

28

2.5 Codificação Aritmética



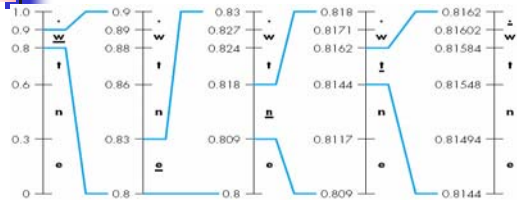
29

2.5 Codificação Aritmética

- Nesse exemplo, o código pode ser qualquer número entre 0,81602 e 0,8162.
 - 0,8161, por exemplo.
- Decodificador conhece o alfabeto, as probabilidades e os intervalos.
 - Então pode seguir o mesmo processo do codificador para decodificar a mensagem 0,8161.

30

2.5 Codificação Aritmética



0,8161 => primeiro caracter é "w", pois 0,8161 está no intervalo 0,8-0,9. O segundo é "e", pois 0,8161 está no intervalo 0,8-0,83. E assim por diante.

31

2.5 Codificação Aritmética

- Nesse método, o número de dígitos no código cresce linearmente de acordo com o tamanho da string.
- Logo, o número máximo de caracteres em uma string é determinado pela precisão de ponto flutuante na máquina destino.
 - Strings grandes podem ser quebradas em duas ou mais substrings.

32

Para Saber Mais

- Halsall, F. Multimedia Communications: Applications, Networks, Protocols, and Standards, Addison-Wesley Publishing, 2001. ISBN: 0201398184. Capítulo 3.
- Gonzales, R.; Woods, R. E. Digital Image Processing. Segunda edição, 2002. Capítulo 8.

33