FEUP Universidade do Porto
Faculdade de Engenharia

Departamento de Engenharia Electrotécnica e de Computadores

Apontamentos de

Teoria da Informação

Sílvio A. Abrantes

2003

1. Shannon e a Teoria da Informação

1.1. Introdução e perspectiva histórica

1.2. Medidas de informação: incerteza, entropia, informação mútua média e capacidade de canal

1.3. Teorema da codificação de canal: ritmo de informação, teorema da codificação de canal, codificação de fonte, redundância

1.4. Teorema da codificação de fonte: códigos unicamente descodificáveis, códigos de prefixação, desigualdade de Kraft-McMillan, teorema da codificação de fonte, extensões

1.5. Códigos de fonte para compressão de dados: técnicas de compressão sem perdas, códigos baseados em modelos estatísticos e em dicionários: código de Shannon-Fano, código de Huffman, código aritmético, código de Lempel-Ziv (L-Z)

1.6. Fontes e canais contínuos: entropia diferencial, teorema da capacidade de canal (lei de Shannon-Hartley), limite de Shannon, plano da probabilidade de erro e plano da eficiência em largura de banda

2. Codificação para correcção de erros

2.1. Introdução

2.2. Técnicas ARQ

2.3. Códigos de blocos

2.4. Códigos cíclicos

2.5. Códigos convolucionais

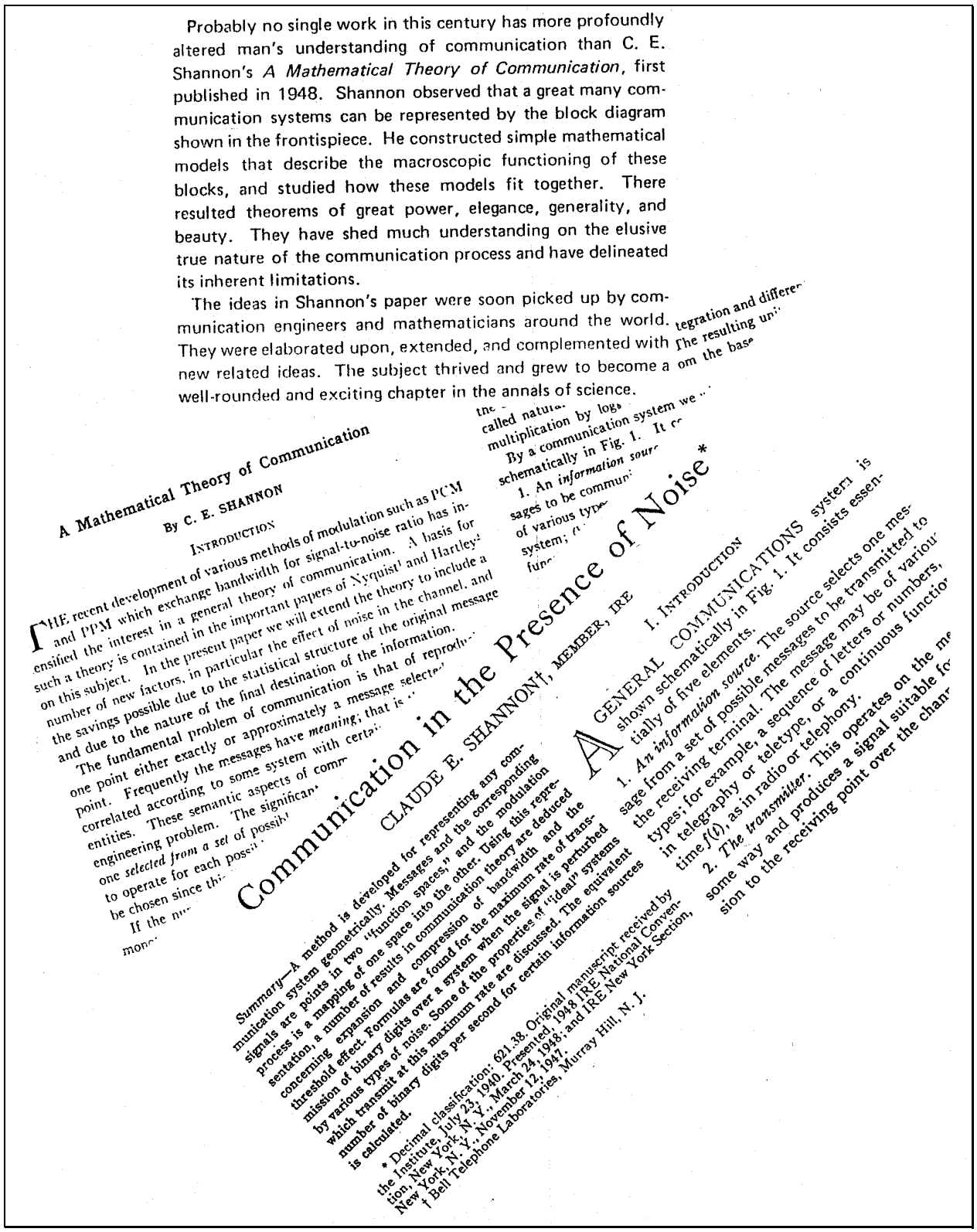
2.6. Códigos entrelaçados e concatenados

1.1. Introdução e perspectiva histórica

Shannon e a Teoria da Informação

Probably no single work in this century has more profoundly altered man's understanding of communication than C. E. Shannon's *A Mathematical Theory of Communication*, first published in 1948. Shannon observed that a great many communication systems can be represented by the block diagram shown in the frontispiece. He constructed simple mathematical models that describe the macroscopic functioning of these blocks, and studied how these models fit together. There resulted theorems of great power, elegance, generality, and beauty. They have shed much understanding on the elusive true nature of the communication process and have delineated its inherent limitations.

The ideas in Shannon's paper were soon picked up by communication engineers and mathematicians around the world. They were elaborated upon, extended, and complemented with new related ideas. The subject thrived and grew to become a well-rounded and exciting chapter in the annals of science.



A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist and Hartley on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

If the number of messages in the set is finite then this number or any monotonic function of this number can be regarded as a measure of the information produced when one message is chosen from the set, all choices being equally likely. As was pointed out by Hartley the most natural choice is the logarithmic function. Although this definition must be generalized considerably when we consider the influence of the statistics of the message and when we have a continuous range of messages, we will in all cases use an essentially logarithmic measure.

The logarithmic measure is more convenient for various reasons:

1. It is practically more useful. Parameters of engineering importance such as time, bandwidth, number of relays, etc., tend to vary linearly with the logarithm of the number of possibilities. For example, adding one relay to a group doubles the number of possible states of the relays. It adds 1 to the base 2 logarithm of this number. Doubling the time roughly squares the number of possible messages, or doubles the logarithm, etc.

2. It is nearer to our intuitive feeling as to the proper measure. This is closely related to (1) since we intuitively measure entities by linear comparison with common standards. One feels, for example, that two punched cards should have twice the capacity of one for information storage, and two identical channels twice the capacity of one for transmitting information.

3. It is mathematically more suitable. Many of the limiting operations are simple in terms of the logarithm but would require clumsy restatement in terms of the number of possibilities.

The choice of a logarithmic base corresponds to the choice of a unit for measuring information. If the base 2 is used the resulting units may be called binary digits, or more briefly *bits*, a word suggested by J. W. Tukey. A device with two stable positions, such as a relay or a flip-flop circuit, can store one bit of information. N such devices can store N bits, since the total number of possible states is 2^N and $\log_2 2^N = N$. If the base 10 is used the units may be called decimal digits. Since

$$\log_2 M = \log_{10} M / \log_{10} 2 = 3.32 \log_{10} M$$

a decimal digit is about 3½ bits. A digit wheel on a desk computing machine has ten stable positions and therefore has a storage capacity of one decimal digit. In analytical work where integration and differentiation are involved the base e is sometimes useful. The resulting units of information will be called natural units. Change from the base a to base b merely requires multiplication by $\log_b a$.

By a communication system we will mean a system of the type indicated schematically in Fig. 1. It consists of essentially five parts:

1. An *information source* which produces a message or sequence of messages to be communicated to the receiving terminal. The message may be of various types: (a) A sequence of letters as in a telegraph of teletype system; (b) A single function of time $f(t)$ as in radio or telephony; (c) A function of time and other variables as in black and white television—here the message may be thought of as a function $f(x, y, t)$ of two space coordinates and time, the light intensity at point (x, y) and time t on a pickup tube plate; (d) Two or more functions of time, say $f(t)$, $g(t)$, $h(t)$—this is the case in "three-dimensional" sound transmission or if the system is intended to service several individual channels in multiplex; (e) Several functions of several variables—in color television the message consists of three functions $f(x, y, t)$, $g(x, y, t)$, $h(x, y, t)$ defined in a three-dimensional continuum—we may also think of these three functions as components of a vector field defined in the region—similarly, several black and white television sources would produce "messages" consisting of a number of functions of three variables; (f) Various combinations also occur, for example in television with an associated audio channel.

2. *The transmitter* which operates on the message in some way and produces a signal suitable for transmission over the channel. In telephony this operation consists merely of changing sound pressure into a proportional electrical current. In telegraphy we have an encoding operation which produces a sequence of dots, dashes and spaces on the channel corresponding to the message. In a multiplex PCM system the different speech functions must be sampled, compressed, quantized and encoded, and finally interleaved properly to construct the signal. Vocoder systems, television and frequency modulation are other examples of complex operations applied to the message to obtain the signal.

Communication in the Presence of Noise*

CLAUDE E. SHANNON†, MEMBER, IRE

Summary—A method is developed for representing any communication system geometrically. Messages and the corresponding signals are points in two "function spaces," and the modulation process is a mapping of one space into the other. Using this representation, a number of results in communication theory are deduced concerning expansion and compression of bandwidth and the threshold effect. Formulas are found for the maximum rate of transmission of binary digits over a system when the signal is perturbed by various types of noise. Some of the properties of "ideal" systems which transmit at this maximum rate are discussed. The equivalent number of binary digits per second for certain information sources is calculated.

I. INTRODUCTION

A GENERAL COMMUNICATIONS system is shown schematically in Fig. 1. It consists essentially of five elements.

1. *An information source.* The source selects one message from a set of possible messages to be transmitted to the receiving terminal. The message may be of various types; for example, a sequence of letters or numbers, as in telegraphy or teletype, or a continuous function of time $f(t)$, as in radio or telephony.

2. *The transmitter.* This operates on the message in some way and produces a signal suitable for transmission over the channel to the receiving point over the chan-

* Decimal classification: 621.38. Original manuscript received by the Institute, July 23, 1940. Presented, 1948 IRE National Convention, New York, N. Y., March 24, 1948; and IRE New York Section, New York, N. Y., November 12, 1947.

† Bell Telephone Laboratories, Murray Hill, N. J.

Introdução

Uma *teoria geral da comunicação* é necessária para adquirirmos uma *visão mais ampla*, uma *perspectiva mais vasta*, que conduza aos princípios básicos do projecto e comparação de sistemas de comunicações.

A **Teoria da Informação** é uma disciplina centrada à volta de uma abordagem matemática comum ao estudo do *armazenamento* e *manipulação* da *informação*. Como tal, fornece uma *base teórica* para actividades como:

- observação, medida, compressão e armazenamento de dados
- telecomunicações
- estimação
- tomada de decisões
- reconhecimento de padrões.

Relativamente às telecomunicações, a Teoria da Informação

- fornece-nos *pistas* para melhorar a *eficiência* da comunicação, com base no estudo das possibilidades e limitações inerentes às leis físicas.

  A **Codificação** (de fonte e de canal) é uma forma de melhorar a eficiência da comunicação.

- *estabelece limites* para essa eficiência, em relação aos quais poderemos comparar diversos sistemas.

Introdução

A Teoria da Informação é uma teoria *matemática* que trata de três conceitos básicos:

- A *medida da informação*.
- A *capacidade* de um canal de comunicações transferir informação.
- A *codificação*, como meio de utilizar os canais com toda a sua capacidade.

Estes três conceitos estão ligados pelo que se pode chamar o

Teorema Fundamental da Teoria da Informação (de Shannon):

Dada uma fonte de informação e um canal de comunicação, existe uma técnica de codificação tal que a informação pode ser transmitida através do canal a qualquer ritmo inferior à capacidade do canal e com uma frequência de erros arbitrariamente pequena apesar da presença do ruído.

- O aspecto *surpreendente* deste teorema é a transmissão *sem erros* através de um canal ruidoso, uma condição que é obtida através de codificação adequada.
- A *codificação* é usada para *adaptar* a fonte e o canal para a máxima transferência fiável de informação.

Introdução

A Teoria da Informação tenta responder a perguntas do género:

- *O que é a informação? Como a medimos?*
- *Quais são os limites fundamentais à transmissão de informação?*
- *Quais são os limites fundamentais à extracção de informação do meio ambiente?*
- *Como é que se devem projectar dispositivos ou equipamentos que se aproximem desses limites?*
- *Os dispositivos e equipamentos actuais aproximam-se, ou não, desses limites?*

Perspectiva histórica

- Prenúncios remotos da Teoria da Informação: **código Morse** (Samuel Morse, 1832-1838).

- As primeiras tentativas para definir uma medida de informação foram feitas pelos teóricos da comunicação **H. Nyquist** (telegrafia e critérios de Nyquist — 1924-1928) e **Hartley** (1928) e pelo estatístico **R. Fisher** (1925).

- **Hartley** definiu pela primeira vez a *quantidade de informação* como *logaritmo* do número de símbolos ou mensagens.

- O assunto só tomou a sua forma actual após os trabalhos de **Claude Shannon** (1948), **Wiener** (1948) e **Kotelnikov** (1947). Dos três foi **Shannon** quem estabeleceu os fundamentos mais importantes e genéricos da teoria, no seu artigo *"The Mathematical Theory of Communication"* (Bell System Technical Journal, Julho '48 e Outubro '48), considerado a *base* da moderna teoria da comunicação.

- Durante a 2ª Guerra Mundial:
  * **N. Wiener** (U.S.A.) e **A. Kolmogoroff** (U.R.S.S.)
    Apresentaram independentemente a solução para alguns problemas relacionados com a influência do ruído sobre os sinais de radar, de modo que estes fossem convenientemente detectados com vista à exacta localização dos aviões inimigos.
  * **Claude Shannon** (U.S.A.) — criptografia
    Tendo começado por considerar as vantagens relativas de muitos sistemas de comunicações novos na época (PCM…), procurou um método geral de comparação dos seus méritos. Revelando uma percepção notável baseada largamente em considerações heurísticas, Shannon procurou uma teoria básica para a comunicação através de canais ruidosos.

- Outras contribuições: **Hamming** (1950), **Feinstein** (1954), **Fano** (1952, 1961), **Elias** (1954, 1963), **Berlekamp** (1964), **Gallager** (1965), **Viterbi** (1967), **Massey** (1969), **Cover** (1972), **Slepian** & **Wolf** (1973), **Blahut** (1977).

Perspectiva histórica

Norbert Wiener ("Cybernetics: Control and Communication in the Animal and the Machine", 1948) colocou a seguinte questão:

Dado um conjunto de sinais possíveis que não são da nossa escolha, mais o ruído (inevitável), como é que faremos a melhor estimativa dos valores presentes e futuros do sinal recebido?

Estamos no âmbito da *Teoria da Detecção*, onde as soluções óptimas para este problema e outros semelhantes são procuradas.

O trabalho de **Shannon** está mais relacionado com o que habitualmente achamos que é a *comunicação*, onde o *processamento do sinal* se pode fazer *tanto no emissor como no receptor*.

Shannon colocou esta questão:

Dado o conjunto de mensagens possíveis que uma fonte pode produzir e que não são da nossa escolha, como é que as mensagens deverão ser representadas para que melhor transmitam a informação num dado sistema com as suas limitações físicas inerentes?

Para tratar deste problema em termos genéricos é necessário concentrarmo-nos mais na informação que nos sinais

1.2. Medidas de informação

- Incerteza
- Entropia
- Informação mútua média
- Capacidade de canal

Medida da informação

Informação — tudo aquilo que é produzido por uma fonte para ser transferido para o utilizador.

No que respeita à medida da informação podemos considerar **dois pontos de vista**:

- UTILIZADOR — a medida da informação está relacionada com ***incerteza*** (em relação à mensagem a ser transmitida).
- FONTE — a medida da informação é uma indicação da ***liberdade de escolha*** exercida pela fonte ao seleccionar uma mensagem.

  - Se a fonte puder escolher livremente entre muitas mensagens diferentes $\Rightarrow$ o utilizador terá muitas dúvidas em relação à mensagem que vai ser escolhida.
  - Se não houver nenhuma possibilidade de escolha ($\Rightarrow$ só uma mensagem possível) $\Rightarrow$ não há incerteza $\Rightarrow$ não há informação.

Medida da informação

A medida da informação envolve probabilidades

Informação (ou auto-informação):

$$I_A = f(P_A) = \log_b \frac{1}{P_A} = -\log_b P_A \qquad (0 \le P_A \le 1)$$

Se $b = 2 \Rightarrow$ bit

$b = e \Rightarrow$ nat

$b = 10 \Rightarrow$ hartley

Um dígito binário (0 e 1) pode transportar uma informação superior ou inferior a 1 bit, consoante as probabilidades de ocorrência de cada dígito.

Esta medida logarítmica de informação é a única função que satisfaz as seguintes propriedades:

1) $I_i \ge 0$ para $0 \le P_i \le 1$ (a auto-informação é não-negativa)

2) $I_i \to 0$ para $P_i \to 1$ (a incerteza aumenta a informação)

3) $I_i > I_j$ para $P_i < P_j$

4) com mensagens independentes x_i e y_j ($\Rightarrow P(x_i y_j) = P_i P_j$)

$$I_{ij} = \log_2 \frac{1}{P_i P_j} = \log_2 \frac{1}{P_i} + \log_2 \frac{1}{P_j} = I_i + I_j$$

(a informação total é a soma das informações individuais)

EXEMPLO: $P_0 = \frac{1}{4}; \qquad P_1 = \frac{3}{4}$

Informação transportada pelo dígito 0: $I_0 = -\log_2 \frac{1}{4} = 2$ bits

Informação transportada pelo dígito 1: $I_1 = -\log_2 \frac{3}{4} = 0,41$ bits

Entropia (ou informação média)

- Auto-informação — é definida em termos das mensagens ou símbolos individuais.
- Informação média (**entropia**) — é definida em termos do conjunto das mensagens que a fonte pode produzir.

Seja uma *fonte discreta X* com *M* símbolos diferentes e estatisticamente independentes.

- Quando o símbolo de ordem *j* é transmitido a informação transportada é $I_j = -\log_2 P_j$ bits.
- A informação média associada aos *M* símbolos da fonte *X* é a **média ponderada** das auto-informações de cada símbolo. A essa informação média por símbolo da fonte chama-se **entropia** e designa-se por *H*(*X*):

$$H(X) = \sum_{j=1}^{M} P_j I_j = -\sum_{j=1}^{M} P_j \log_2 P_j \quad \text{bits/símbolo}$$

O que é que significa a **entropia** de uma fonte[1]? Significa que:

> *Embora não possamos prever qual o símbolo que a fonte irá produzir a seguir, em média esperamos obter H bits de informação por símbolo, ou NH bits numa mensagem de N símbolos, se N for elevado.*

[1]Em *Física Estatística* a entropia de um sistema físico é definida por $S = k \ln \Omega$, em que *k* é a constante de Boltzmann e Ω o número de estados acessíveis ao sistema. Esta função, que fornece uma medida quantitativa do grau de *casualidade* do sistema, é semelhante na forma à entropia da Teoria da Informação.

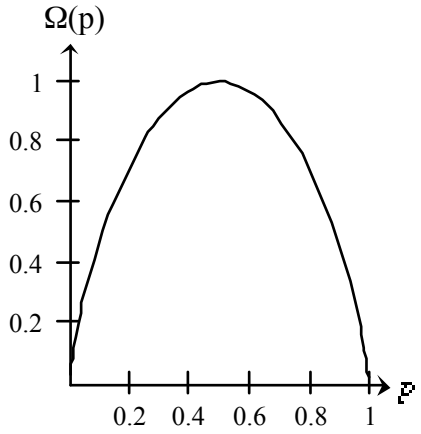
Entropia

- Vamos reescrever a fórmula da entropia como

$$H(P_1, P_2, \ldots, P_M) = -\sum_{j=1}^{M} P_j \log_2 P_j \qquad \text{(bits/símbolo)}$$

- No caso de uma fonte ou mensagem binária com 2 saídas possíveis, com probabilidades *p* e *1-p*, a entropia é designada por $\Omega(p)$ e vale

$$\Omega(p) = H(p, 1-p) = -p \log_2 p - (1-p)\log_2(1-p)$$



Entropia de uma fonte binária

- De modo geral, com um alfabeto de **M** símbolos tem-se

$$O \le H(X) \le \log_2 M$$

↓ ↓

A fonte não fornece informação, em média ⇒ não há incerteza quanto à mensagem.

Ex.: a fonte produz continuamente o mesmo símbolo (todas as probabilidades dos símbolos são nulas excepto uma, unitária)

Máxima incerteza ou máxima liberdade de escolha ⇒ todos os símbolos são equiprováveis (nenhum é favorecido).

Ex.:

$$P_j = \frac{1}{M} \Rightarrow H = H_{\max} =$$
$$= P_1 I_1 + P_2 I_2 + \ldots = M P_j I_j = \log_2 M$$

Entropia — Exemplos

EXEMPLO 1

Qual é a informação média, em bits/carácter, da língua portuguesa (23 letras) supondo cada letra equiprovável?

R.: $H = -\log\frac{1}{23} = 4,52$ bits /carácter ♦ *Limite Superior*

EXEMPLO 2

Na realidade as letras não são equiprováveis. Admitindo (ADMITINDO...) que as letras ocorrem com as probabilidades indicadas, quanto vale a entropia?

A, E, O, T:	p = 0,10
H, I, N, R, S:	p = 0,07
C, D, F, L, M, P, U:	p = 0,02
B, G, J, Q, V, X, Z:	p = 0,01

R.:

$$H = -(4\times 0,10\log_2 0,1 + 5\times 0,07\log_2 0,07 + 7\times 0,02\log_2 0,02 + 7\times 0,01\log_2 0,01) =$$
$$= 3,92 \text{ bits/carácter}$$

Entropia máxima

Teorema

A entropia $H(X)$ de uma fonte respeita sempre $0 \le H(X) \le \log_2 M$. O valor máximo atinge-se apenas quando todas as saídas da fonte são equiprováveis.

Demonstração

A demonstração baseia-se na desigualdade $\ln x \le x - 1$. Seja $\nu = \frac{Q_i}{P_i}$, em que Q_j e P_i são probabilidades discretas válidas, isto é,

$$P_i, Q_i \ge 0 \quad \text{e} \quad \sum_{i=1}^{M} P_i = \sum_{i=1}^{M} Q_i = 1$$

Sendo então $\ln x \le x - 1$, teremos:

$$\ln \frac{Q_i}{P_i} \le \frac{Q_i}{P_i} - 1 \text{, ou } -P_i \ln P_i \le Q_i - P_i - P_i \ln Q_i$$

Somando para todos os *ii*:

$$-\sum_{i=1}^{M} P_i \ln P_i \le \sum_{i=1}^{M} Q_i - \sum_{i=1}^{M} P_i - \sum_{i=1}^{M} P_i \ln Q_i$$

Mas $\sum_{i=1}^{M} Q_i - \sum_{i=1}^{M} P_i = 1 - 1 = 0$ e $\ln z = \ln 2 \log_2 z$. Portanto,

$$-\ln 2 \sum_{i=1}^{M} P_i \log_2 P_i \le -\ln 2 \sum_{i=1}^{M} P_i \log_2 Q_i \qquad \Rightarrow \qquad H(X) \le -\sum_{i=1}^{M} P_i \log_2 Q_i$$

Isto é válido para todos os valores de Q_j, incluindo $Q_i = \frac{1}{M}$ (porque não?).

$$\Rightarrow \qquad H(X) \le -\sum_{i=1}^{M} P_i \log_2 \frac{1}{M} = \log_2 M \sum_{i=1}^{M} P_i = \log_2 M \,. \qquad \textit{c.q.d.}$$

Exemplo: informação contida numa imagem de televisão e em texto

Televisão (com Independência entre Símbolos)

- Linhas activas: 575
- Pontos de ecrã: 350 000 $\Rightarrow$ 608 pontos/linha $\Rightarrow$ 608 × 575 = 349 600 pontos
- Informação média (entropia) máxima:

  (Preto e Branco, *m* = 8 níveis de luminância por "spot" (níveis equiprováveis))

  $$-\log_2 1/8 = 3 \text{ bits}$$

  (A Cores (vermelho, verde, azul))

  $$\Rightarrow m = 8 \times 3 = 24 \Rightarrow \text{entropia máxima: } -\log_2 1/24 = 4,58 \text{ bits}$$

- Em cada imagem: Preto & Branco: 3×349 600 = 1 048 800 bits/imagem

  A cores: 4,58 × 349 600 = 1 601 973 bits/imagem

- Projecção da imagem (50 quadros/s $\Rightarrow$ 25 imagens/s)

  Preto e Branco: 25 × 1 048 800 = 26 220 000 bit/s

  A cores: 25 × 1 601 973 = 40 049 312 bits/s

Texto (letras equiprováveis)

Português: 23 letras + espaço → auto-informação máxima : $-\log \frac{1}{24} = 4,58$ bits

- comprimento médio de cada palavra = 23
- Conteúdo de informação por palavra (entropia): 23 × 4,58 = 105,4 bits/palavra

  Comparando com uma imagem a Preto e Branco, uma descrição impressa necessitaria de $\frac{1048800}{105,4} \approx 9951$ *palavras.*

Russo: 33 letras + espaço → 33 log 34 = 33 × 5,1 = 168,3 bits/palavra $\Rightarrow$ *6232 palavras*

Inglês: 26 letras + espaço → 26 log 27 = 26 × 4,8 = 124,8 bits/palavra $\Rightarrow$ *8404 palavras*

Caracterização de canais

Consideremos um canal com *M* entradas seleccionadas de um alfabeto X e *N* saídas seleccionadas de um alfabeto Y. Podemos encarar a saída do canal como a versão ruidosa da entrada.

É costume caracterizar o canal de duas maneiras:

- através do *diagrama de transição*
- através da *matriz de probabilidades de transição* (ou *matriz do canal*)

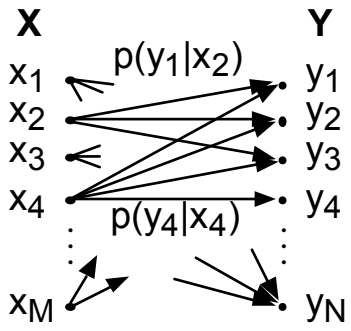


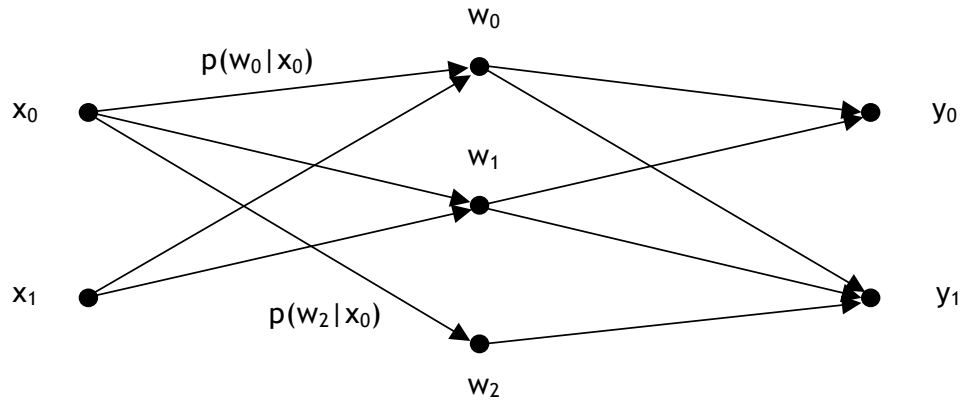
Diagrama de transição do canal

$$[P(Y|X)] = \begin{bmatrix} p(y_1|x_1) & p(y_2|x_1) & \dots & p(y_N|x_1) \\ p(y_1|x_2) & p(y_2|x_2) & \dots & p(y_N|x_2) \\ \dots & \dots & \dots & \dots \\ p(y_1|x_M) & p(y_2|x_M) & \dots & p(y_N|x_M) \end{bmatrix}$$

Matriz de probabilidades de transição do canal

- Nesta matriz cada linha está associada ao correspondente símbolo de entrada e cada coluna está associada ao correspondente símbolo de saída.
- A soma dos elementos de cada linha é 1: $\sum_j p(y_j|x_i) = 1$

Simplificação de canais discretos compostos



- **A partir das matrizes de transição:**

  A matriz de transição global é igual ao produto das matrizes de transição individuais.

  Exemplo:

$$\left[P(W \mid X)\right] = \begin{bmatrix} 0,2 & 0,3 & 0,5 \\ 0,4 & 0,5 & 0,1 \end{bmatrix} \qquad \left[P(Y \mid W)\right] = \begin{bmatrix} 0,6 & 0,4 \\ 0,5 & 0,5 \\ 0,7 & 0,3 \end{bmatrix}$$

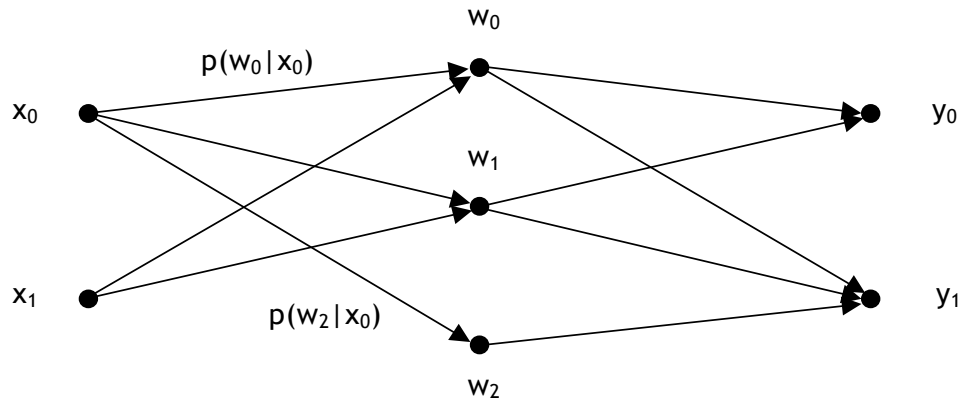
$$\Rightarrow \quad \left[P(Y \mid X)\right] = \left[P(W \mid X)\right] \times \left[P(Y \mid W)\right] = \begin{bmatrix} 0,62 & 0,38 \\ 0,56 & 0,44 \end{bmatrix}$$

- **A partir de um diagrama de probabilidades de transição:**

  As probabilidades condicionais $p(y_j \mid x_i)$ (**x_i** – entrada; **y_j** – saída) são iguais à soma das probabilidades associadas aos diversos "trajectos" de x_i para y_j.

Simplificação de canais discretos compostos: exemplo

Exemplo com o diagrama de probabilidades de transição: como simplificar o canal composto seguinte?



- Há dois percursos de x_0 para y_0: $x_0 \rightarrow w_0 \rightarrow y_0$ e $x_0 \rightarrow w_1 \rightarrow y_0$

$$\Rightarrow P(y_0 \mid x_0) = \underbrace{P(w_0 \mid x_0)P(y_0 \mid w_0)}_{x_0 \rightarrow w_0 \rightarrow y_0} + \underbrace{P(w_1 \mid x_0)P(y_0 \mid w_1)}_{x_0 \rightarrow w_1 \rightarrow y_0}$$

- Há dois percursos de x_1 para y_0: $x_1 \rightarrow w_0 \rightarrow y_0$ e $x_1 \rightarrow w_1 \rightarrow y_0$

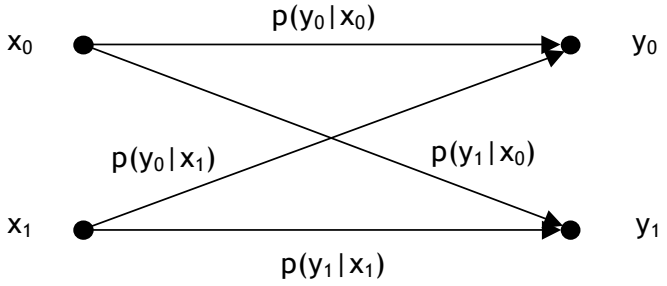
$$\Rightarrow \quad P(y_0 \mid x_1) = P(w_0 \mid x_1)P(y_0 \mid w_0) + P(w_1 \mid x_1)P(y_0 \mid w_1)$$

- Há três percursos de x_0 para y_1:

  $x_0 \rightarrow w_0 \rightarrow y_1$, $x_0 \rightarrow w_1 \rightarrow y_1$ e $x_0 \rightarrow w_2 \rightarrow y_1$

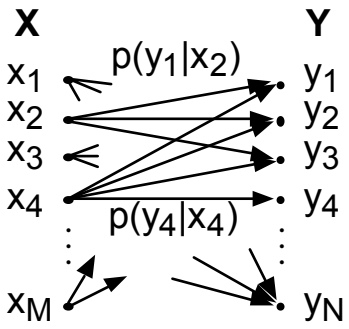
  $P(y_1 \mid x_0)$ é a soma das probabilidades associadas aos três percursos, etc.

O resultado final seria o canal binário seguinte:



Entropia Condicional

Admitamos que há dois acontecimentos, *X* e *Y*, com *M* possibilidades para *X* e *N* para *Y*. Seja $P(x_i, y_j)$ a probabilidade conjunta da ocorrência (x_i, y_j) e $P(y_j \mid x_i)$ a probabilidade condicional de y_j ocorrer dado x_i ter ocorrido.



- A entropia condicional de *Y* dada a ocorrência x_i é definida como

$$H(Y \mid x_i) = \sum_{j=1}^{N} P(y_j \mid x_i) \log_2 \frac{1}{P(y_j \mid x_i)} = -\sum_{j=1}^{N} P(y_j \mid x_i) \log_2 P(y_j \mid x_i) =$$
$$= H\left[P(y_1 \mid x_i), P(y_2 \mid x_i), \ldots, P(y_N \mid x_i) \right]$$

As probabilidades condicionais envolvidas são as dos ramos que saem de x_i.

- A entropia condicional de *Y* dado *X* é definida como a média ponderada de $H(Y \mid x_i)$ para todos os valores de x_i:

$$H(Y \mid X) = \sum_{i=1}^{M} P(x_i) H(Y \mid x_i) =$$
$$= -\sum_{i=1}^{M} \sum_{j=1}^{N} P(x_i) P(y_j \mid x_i) \log P(y_j \mid x_i)$$

entropia condicional

- A entropia conjunta de *X* e *Y* é dada por

$$H(X,Y) = \sum_{i=1}^{M} \sum_{j=1}^{N} P(x_i, y_j) \log_2 \frac{1}{P(x_i, y_j)}$$

entropia conjunta

- Prova-se que $H(X,Y) \leq H(X) + H(Y)$.

Entropia Condicional

P.: *Qual é a relação entre a entropia conjunta* $H(X,Y)$ *e a entropia condicional* $H(Y|X)$*?*

R.: Como $P(x_i, y_j) = P(x_i)P(y_j \mid x_i)$ (regra de Bayes) então

$$H(X,Y) = \sum_{i=1}^{M}\sum_{j=1}^{N} P(x_i)P(y_j \mid x_i)\log_2 \frac{1}{P(x_i)P(y_j \mid x_i)} =$$
$$= -\sum_{i=1}^{M}\sum_{j=1}^{N} P(x_i)P(y_j \mid x_i)\log_2 P(x_i) - \sum_{i=1}^{M}\sum_{j=1}^{N} P(x_i)P(y_j \mid x_i)\log_2 P(y_j \mid x_i)$$

Mas

- $$\sum_{j=1}^{N} P(x_i)P(y_j \mid x_i)\log_2 P(x_i) = P(x_i)\log_2 P(x_i)\underbrace{\sum_{j=1}^{N} P(y_j \mid x_i)}_{1} = P(x_i)\log_2 P(x_i)$$

- $$-\sum_{i=1}^{M}\sum_{j=1}^{N} P(x_i)P(y_j \mid x_i)\log_2 P(y_j \mid x_i) = \sum_{i=1}^{M} P(x_i)H(Y \mid x_i) = H(Y \mid X)$$

$$\Rightarrow \qquad H(X,Y) = -\sum_{i=1}^{M} P(x_i)\log_2 P(x_i) + H(Y \mid X) =$$
$$= H(X) + H(Y \mid X)$$

- Se os símbolos forem independentes $\Rightarrow H(X,Y) = H(X) + H(Y)$
- Do mesmo modo se provaria que $H(X,Y) = H(Y) + H(X \mid Y)$.
- Como $H(X,Y) \le H(X) + H(Y) \quad \Rightarrow \quad H(X \mid Y) \le H(X)$ e $H(Y \mid X) \le H(Y)$
- Como se vê, a entropia de símbolos dependentes é menor que a entropia de símbolos independentes.

Entropia condicional: um exemplo

Uma fonte produz três símbolos, A, B e C, com as seguintes probabilidades:

$$\{P(X=A), P(X=B), P(X=C)\} = \left\{\frac{9}{27}, \frac{16}{27}, \frac{2}{27}\right\}$$

	$P(y_j \mid x_i)$	y_j = A	B	C
	A	0	4/5	1/5
x_i	B	1/2	1/2	0
	C	1/2	2/5	1/10

***P.:** Determine a entropia da fonte, $H(X)$, e as entropias $H(Y \mid X)$ e $H(X,Y)$.*

R.: As probabilidades conjuntas $P(x_i, y_j) = P(x_i)P(y_j \mid x_i)$ valem:

	$P(x_i,y_j)$	y_j = A	B	C
	A	0	4/15	1/15
x_i	B	8/27	8/27	0
	C	1/27	4/135	1/135

Como $P(y_j) = \sum_i P(x_i, y_j) \Rightarrow \quad \{P(Y=A), P(Y=B), P(Y=C)\} = \left\{\frac{1}{3}, \frac{16}{27}, \frac{2}{27}\right\}$

Assim, $H(Y) = H(1/3, 16/27, 2/27) = 1,29$ bits/símbolo (= $H(X)$)

A entropia condicional $H(Y \mid X)$ será menor. De facto:

$$H(Y \mid X=A) = H(0, 4/5, 1/5) = 0,72$$

$$H(Y \mid X=B) = H(1/2, 1/2, 0) = 1$$

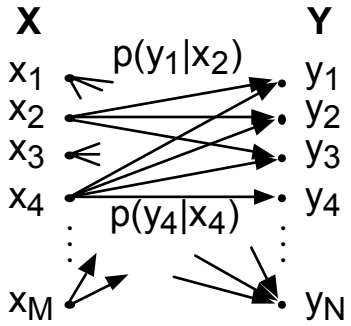
$$H(Y \mid X=C) = H(1/2, 2/5, 1/10) = 1,36$$

$$H(Y \mid X) = \frac{9}{27} \times 0,72 + \frac{16}{27} \times 1 + \frac{2}{27} \times 1,36 = 0,93 \text{ bits/símbolo}$$

Portanto, a *entropia conjunta* vale $H(X,Y) = 1,29 + 0,93 = 2,22$ bits/símbolo.

Equivocação em canais discretos com ruído

Consideremos o canal discreto do diagrama seguinte:



- A *entropia da fonte* de mensagens, *H(X)*, depende apenas da fonte X. *É a incerteza em relação a x_i*, isto é, em relação a que símbolo será transmitido.
- A *entropia da saída* do canal, *H(Y)* — que é equivalente a uma nova fonte — depende da fonte X e dos erros de transmissão. *É a incerteza em relação ao símbolo que será recebido*, no caso de uma dada fonte de mensagens e de um dado canal de comunicação.
- Conhecendo as estatísticas da fonte de mensagens e do canal ruidoso conhece-se a *entropia conjunta, H(X,Y). É a incerteza sobre a transmissão de x_i e a recepção de y_j.*
- Suponhamos que conhecemos x_i (isto é, sabemos que símbolo foi enviado):

$$H(Y \mid X) = -\sum_{i=1}^{M}\sum_{j=1}^{M} p(x_i)\,p(y_j \mid x_i)\log_2 p(y_j \mid x_i) \qquad \textit{Entropia condicional}$$

  É a incerteza de receber y_j quando x_i é enviado, isto é, é a incerteza média do emissor em relação ao que será recebido.

- Suponhamos que conhecemos y_j (isto é, sabemos que símbolo foi recebido):

$$H(X \mid Y) = -\sum_{j=1}^{M}\sum_{i=1}^{M} p(y_j)\,p(x_i \mid y_j)\log_2 p(x_i \mid y_j) \quad \textit{Entropia condicional.}$$

  É a incerteza sobre x_i ter sido enviado quando se recebe y_j, ou a incerteza média do receptor da mensagem em relação ao que foi realmente enviado. Por outras palavras, é a incerteza sobre a entrada que resta depois da saída ter sido observada.

  A $H(X \mid Y)$ chama-se *equivocação do canal*: é a medida da informação *perdida* devido ao canal.

Informação mútua média

- A entropia $H(X)$ representa a incerteza "a priori" sobre a entrada do canal **antes** de observarmos a saída.
- A entropia condicional $H(X|Y)$ representa a incerteza "a posteriori" sobre a entrada do canal **depois** de observarmos a saída. É a incerteza remanescente sobre o acontecimento X após se conhecer o acontecimento Y.
- A diferença $H(X) - H(X|Y)$ deve então ser a incerteza sobre a entrada que eliminamos observando a saída.

  Quer dizer que $H(X) - H(X|Y)$ traduz a *redução de incerteza* sobre X que ganhamos pela observação de Y.

  Por exemplo, se depois da observação de Y continuarmos exactamente com as mesmas dúvidas que tínhamos sobre a entrada X, então não reduzimos incerteza nenhuma: $H(X) - H(X|Y) = 0$.

- À diferença $H(X) - H(X|Y)$ chama-se *informação mútua média* entre a entrada X e a saída Y. Pode ser interpretada como um ganho de informação sobre X, que não teríamos se não conhecêssemos Y.

$$I(X;Y) = H(X) - H(X|Y)$$

- Quanto maior for $H(X|Y)$ menor é $I(X;Y)$. Daí se lhe chamar *equivocação*.
- Uma determinada ocorrência y_j pode tornar a decisão sobre a ocorrência x_i ainda mais incerta ($I(x_i;y_j) < 0$); o conhecimento da distribuição de probabilidades de Y ajuda a reduzir essa incerteza ($I(X;Y) \geq 0$).

Informação mútua média

- *Informação mútua média* entre a entrada X e a saída Y do canal:

$$I(X;Y) = H(X) - H(X \mid Y) =$$
$$= \sum_{i=1}^{M} p(x_i) \frac{1}{\log_2 p(x_i)} - \sum_{i=1}^{M}\sum_{j=1}^{N} p(y_j) p(x_i \mid y_j) \log_2 \frac{1}{p(x_i \mid y_j)}$$

Mas $P(x_i) = \sum_{j=1}^{N} P(x_i, y_j)$, logo

$$I(X;Y) = \sum_{i=1}^{M}\sum_{j=1}^{N} p(x_i, y_j) \left[\log_2 \frac{1}{p(x_i)} + \log_2 \frac{p(x_i, y_j)}{p(y_j)} \right] =$$
$$= \sum_{i=1}^{M}\sum_{j=1}^{N} p(x_i, y_j) \underbrace{\log_2 \frac{p(x_i, y_j)}{p(x_i) p(y_j)}}_{I(x_i, y_j)} =$$
$$= \sum_{i=1}^{M}\sum_{j=1}^{N} p(x_i, y_j) I(x_i, y_j)$$

- $I(x_i, y_j)$ é a informação mútua entre a entrada x_i e a saída y_j,

$$I(x_i, y_j) = \log_2 \frac{p(x_i, y_j)}{p(x_i) p(y_j)} = \log_2 \frac{p(y_j \mid x_i)}{p(y_j)} = \log_2 \frac{p(x_i \mid y_j)}{p(x_i)}.$$

A *informação mútua média* $I(X;Y)$ é a média das informações mútuas entre as diversas entradas e saídas.

- Prova-se também que $I(X;Y) = H(Y) - H(Y \mid X)$.

Informação mútua média: um exemplo

Uma fonte X produz símbolos com probabilidades $\{1/2, 1/3, 1/6\}$, os quais atravessam um canal com a matriz de probabilidades de transição

$$\begin{bmatrix} 2/3 & 0 & 1/3 \\ 1/3 & 2/3 & 0 \\ 0 & 1/3 & 2/3 \end{bmatrix}.$$

A incerteza que temos sobre os símbolos produzidos pela fonte X é mensurável: $H(X) = H(1/2, 1/3, 1/6) = 1,46$ bits/símbolo.

P.: a) De quanto reduzimos a incerteza sobre X observando a saída Y?
b) Calcule a equivocação do canal e a informação mútua $I(x_2, y_1)$.

R.: a) Sendo $P(y_j) = \sum_i P(x_i)P(y_j \mid x_i)$ $\Rightarrow \{P(y_1), P(y_2), P(y_3)\} = \left\{\frac{4}{9}, \frac{5}{18}, \frac{5}{18}\right\}$

$$H(Y) = H(4/9, 5/18, 5/18) = 1,55 \text{ bits/símbolo}$$

$$H(Y \mid X) = \sum_i P(x_i)H(Y \mid x_i) =$$

$$= \frac{1}{2}H\left(\frac{2}{3}, 0, \frac{1}{3}\right) + \frac{1}{3}H\left(\frac{1}{3}, \frac{2}{3}, 0\right) + \frac{1}{6}H\left(0, \frac{1}{3}, \frac{2}{3}\right) = 0,92$$

A informação mútua média vale, portanto, $I(X;Y) = 1,55 - 0,92 = 0,63$ bits/símbolo. Isto significa que a observação da saída do canal baixou a incerteza de $H(X) = 1,46$ para 1,46 - 0,63 = 0,83 bits/símbolo.

b) Como $I(X;Y) = H(Y) - H(Y \mid X) = H(X) - H(X \mid Y)$, concluímos que a equivocação devida ao canal é $H(X \mid Y) = H(X) - I(X;Y) = 0,83$ bits/símbolo.

Será que a observação do símbolo y_1 ajuda a decidir se o símbolo x_2 foi emitido pela fonte? Vamos ver:

$$I(x_2, y_1) = \log_2 \frac{P(y_1 \mid x_2)}{P(y_1)} = \log_2 \frac{3}{4} = -0,42 \text{ bits/símbolo}$$

Conclui-se que esta observação específica aumenta a indefinição sobre x_2.

Informação e entropia: definições e relações importantes

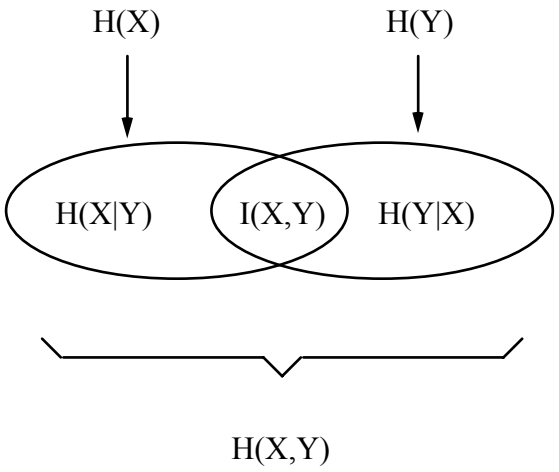
X, Y — variáveis aleatórias com ocorrências x_i e y_j, $i = 1,2,...,M$ e $j = 1,2,...,N$

	↓ *Variável X*	↓ *Variáveis X e Y*
uma ocorrência →	**Auto-informação** $I(x_i) = -\log_2 P(x_i)$	**Informação mútua** $I(x_i;y_j) = \log_2 \frac{P(x_i,y_j)}{P(x_i)P(y_j)}$
várias ocorrências →	**Entropia** (informação média) $H(X) = \sum_{i=1}^{M} P(x_i)I(x_i) = -\sum_{i=1}^{M} P(x_i)\log_2 P(x_i)$	**Informação mútua média** $I(X;Y) = \sum_{i=1}^{M}\sum_{j=1}^{N} P(x_i,y_j)I(x_i;y_j) = \sum_{i=1}^{M}\sum_{j=1}^{N} P(x_i,y_j)\log_2 \frac{P(x_i,y_j)}{P(x_i)P(y_j)}$

- $H(X,Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y)$
- $H(X,Y) \leq H(X) + H(Y)$
- $H(X \mid Y) \leq H(X)$
- $H(Y \mid X) \leq H(Y)$
- $I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$
- $I(X;Y) = H(X) + H(Y) - H(X,Y)$

Informação mútua média, entropias condicionais e diagrama de Venn

É muito fácil de estabelecer as relações entre as diferentes entropias condicionais e entre estas e a informação mútua média se usarmos o seguinte diagrama de Venn:



Repetem-se aqui as relações já conhecidas. Ora confirme-as com o diagrama!

- $H(X,Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y)$
- $H(X,Y) \leq H(X) + H(Y)$
- $H(X \mid Y) \leq H(X)$
- $H(Y \mid X) \leq H(Y)$
- $I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$
- $I(X;Y) = H(X) + H(Y) - H(X,Y)$

Informação mútua média

A informação mútua média é sempre não negativa:

Teorema

*Seja XY o conjunto discreto dos pares (x_i,y_j). A informação mútua média entre X e Y satisfaz I(X;Y) ≥ 0, em que a igualdade só é satisfeita **sse** x e y forem estatisticamente independentes.*

Demonstração

Provemos que *-I(X;Y) ≤ 0*. Sabemos que $\log_2 a = \ln a \log_2 e$:

$$-I(X;Y) = -\log e \sum_i \sum_j P(x_i, y_j) \ln \frac{P(x_i \mid y_j)}{P(x_i)} = \log e \sum_i \sum_j P(x_i, y_j) \ln \frac{P(x_i)}{P(x_i \mid y_j)}$$

Como $\ln \nu \le \nu - 1$:

$$-I(X,Y) \le \log e \sum_i \sum_j P(x_i, y_j) \left[\frac{P(x_i)}{P(x_i \mid y_j)} - 1 \right] = \log e \sum_i \sum_j \left[P(x_i)P(y_j) - P(x_i, y_j) \right] =$$

$$= \log e \left[\underbrace{\sum_i P(x_i)}_{1} \underbrace{\sum_j P(y_j)}_{1} - \underbrace{\sum_i \sum_j P(x_i, y_j)}_{1} \right] = 0 \qquad \Rightarrow -I(X;Y) \le 0$$

Note-se que $I(X,Y) = 0$ ***sse*** $\ln \nu \le \nu - 1 \Rightarrow \nu = 1 \Rightarrow$ ***sse*** $P(x) = P(x \mid y)$ $\Rightarrow$ se ***x*** e ***y*** forem estatisticamente independentes, ***c.q.d.***

Demonstração alternativa:

Sendo $I(X,Y) = H(X) + H(Y) - H(X,Y)$ e $H(X,Y) \le H(X) + H(Y)$, então necessariamente $I(X,Y) \ge 0$. Se *X* e *Y* forem estatisticamente independentes, então $H(X,Y) = H(X) + H(Y) \Rightarrow I(X,Y) = 0$. ***c.q.d.***

Informação mútua média

Como $I(X;Y) \geq 0$ e $I(X;Y) = H(X) - H(X|Y) \Rightarrow$ **$H(X) \geq H(X|Y)$**

(e, tal como antes, H(X) = H(X|Y) **sse** X e Y forem estatisticamente independentes)

Esta desigualdade mostra que se colocarmos restrições a um conjunto X (fazendo-o depender de outras ocorrências) *diminuimos a sua entropia*.

É o que acontece com as línguas (português, francês, etc.).

Por exemplo em inglês (26 letras mais espaço):

- **Codificação** letra a letra, considerando cada letra equiprovável:

  São precisos $-\log_2 27 = 4,76$ dígitos binários/carácter. (entropia máxima)

- **Codificação** letra a letra tendo em conta as probabilidades relativas de ocorrência:

  São precisos 4,03 dígitos binários/carácter. (entropia condicional)

- **Codificação** palavra a palavra tendo em conta a frequência relativa das palavras:

  Só são precisos 1,66 dígitos binários/carácter. (entropia condicional)

Informação mútua média e capacidade de canal

$$I(X;Y) = H(X) - H(X \mid Y)$$

↑ ↑

ENTROPIA DA FONTE — EQUIVOCAÇÃO

É a incerteza média sobre a fonte de informação X

É a incerteza média sobre a fonte depois de se observar a saída do canal, Y

A diferença entre estas duas quantidades é a *informação mútua média* — é uma medida da informação que passou através do canal.

$\Rightarrow$ *Quanto maior for a equivocação menor é a quantidade de informação que passa através do canal.*

Definição:

A *capacidade do canal* é o valor *máximo* da informação mútua média.

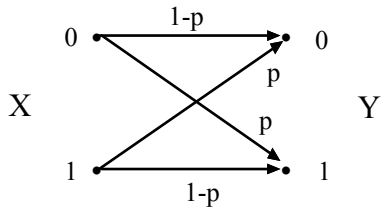
$$C_S = \max_{p_X(x)} I(X;Y) \text{ (bits/símbolo)}$$

Visto o canal ser fixo a maximização deve fazer-se relativamente às probabilidades da fonte.

Exemplo de capacidade de canal

Exemplo 1: Canal binário simétrico ("Binary symmetric channel", BSC)

$P(0|1) = P(1|0) = p$ e $P(0|0) = P(1|1) = 1 - p$.



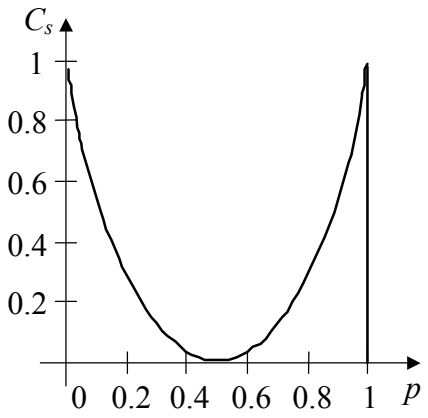
Como $I(X;Y) = H(Y) - H(Y|X)$ vamos calcular estas entropias:

A entropia *H*(*Y*) é máxima quando as saídas são equiprováveis $\Rightarrow$ entradas equiprováveis. Nesse caso *H*(*Y*) = 1 bit/símbolo.

$$H(Y|X) = \sum_{i=1}^{M} P(x_i) H(Y|x_i) = P(X=0)H(Y|X=0) + P(X=1)H(Y|X=1) =$$
$$= P(X=0)\underbrace{H(1-p,p)}_{\Omega(p)} + P(X=1)\underbrace{H(p,1-p)}_{\Omega(p)} = \Omega(p)$$

A capacidade vale, portanto, $C_s = 1 - \Omega(p)$, isto é:

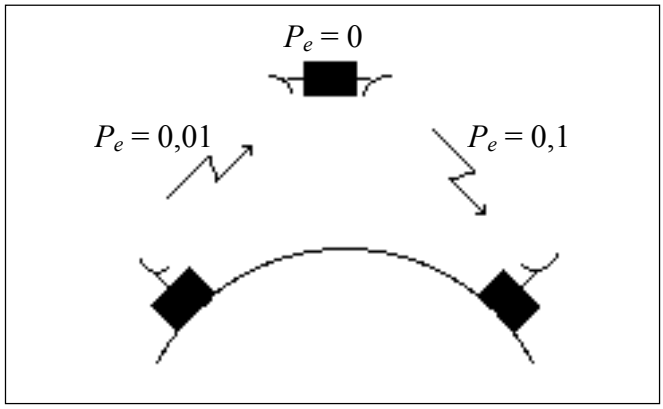
Capacidade do canal binário simétrico $\rightarrow C_s = 1 + p\log_2 p + (1-p)\log_2(1-p)$



Se $P(0|1) = P(1|0) = 0{,}5$, isto é, dada uma determinada saída (1 ou 0) temos 50% de probabilidades de acertarmos na entrada realmente enviada $\Rightarrow$ a capacidade do canal é nula $\Rightarrow$ nesse caso o canal não serve para nada: basta "deitar uma moeda ao ar" no destino.

Exemplo de capacidade de canal

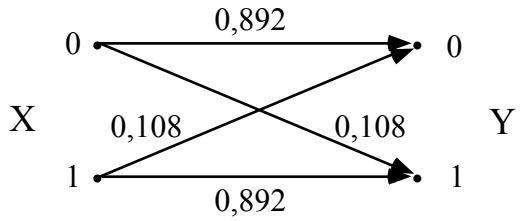
Exemplo 2: Comunicação via satélite



Há 3 canais envolvidos: canal ascendente, canal a bordo e canal descendente. Para determinar a capacidade do sistema global temos de primeiro calcular a matriz de probabilidades de transição global multiplicando as matrizes de probabilidade de transição individuais:

$$[P(Y \mid X)] = \begin{bmatrix} 0{,}99 & 0{,}01 \\ 0{,}01 & 0{,}99 \end{bmatrix} \begin{bmatrix} 1{,}0 & 0 \\ 0 & 1{,}0 \end{bmatrix} \begin{bmatrix} 0{,}9 & 0{,}1 \\ 0{,}1 & 0{,}9 \end{bmatrix} = \begin{bmatrix} 0{,}892 & 0{,}108 \\ 0{,}108 & 0{,}892 \end{bmatrix}$$

O diagrama de transição global deste canal BSC é:



A capacidade do sistema global é, portanto:

$$C_s = 1 + 0{,}892 \log_2 0{,}892 + 0{,}108 \log_2 0{,}108 = 0{,}51 \text{ bit/símbolo}$$

(As probabilidades hipótese são muito mais elevadas que num caso real. Tomaram-se para simplificar os cálculos).

1.3. Teorema da codificação de canal

- Ritmo de informação
- Teorema da codificação de canal

Ritmo (ou taxa) de informação

- Se a fonte de entropia *H(X)* emitir uma sequência de $n \gg 1$ símbolos a *informação total* a ser transferida é de $nH(X)$ bits.

- Se a fonte produzir *r* símbolos/s a *duração* da sequência de *n* símbolos é *n*/*r* segundos.

- Portanto, a informação deve ser transferida à razão média de:

$$R = \frac{\text{informação total}}{\text{duração total}} = \frac{nH(X)}{n/r} = rH(X) \text{ bits/s}$$

Esta é a *taxa de informação* da fonte.

Se a fonte produzir símbolos com diferentes durações τ_i (segundos) o ritmo, ou taxa, de informação pode ser calculado alternativamente como

$$R = \frac{H(X)}{\overline{\tau}} \text{ bits/s}$$

em que $\overline{\tau}$ é a duração média de cada símbolo,

$$\overline{\tau} = P_1\tau_1 + P_2\tau_2 + \ldots + P_M\tau_M = \sum_{i=1}^{M} P_i\tau_i \, .$$

O teorema da codificação de canal
(aplicado a canais discretos sem memória)

A *entropia* representa a *informação esperada* por cada símbolo de fonte.

- Se tivermos duas fontes com iguais entropias, aquela que produzir mais símbolos por unidade de tempo vai colocar mais exigências no sistema de comunicações.

  (imagine duas pessoas a ler o mesmo texto escrito numa língua que domina mal. Uma delas lê depressa. Qual é a que entende melhor?)

  - O *ritmo de informação* mede a quantidade de informação *produzida* num certo intervalo de tempo.

- Existem limitações físicas fundamentais à transferência de informação (ruído, largura de banda, …), que se reflectem na capacidade do canal:

  - A *capacidade do canal* (em bits/s) mede a quantidade de informação que um *canal* pode *transferir* por unidade de tempo.

Shannon relacionou estes dois conceitos através do *teorema da codificação de canal*, também chamado *Teorema Fundamental da Teoria da Informação*:

> **Dado um canal de capacidade C e uma fonte com ritmo de informação R, então se R ≤ C existe uma técnica de codificação tal que a saída da fonte pode ser transmitida através do canal com uma frequência arbitrariamente pequena de erros, apesar da presença de ruído.**
>
> **Se R > C, não é possível a transmissão sem erros.**

O teorema da codificação de canal aplicado a canais discretos sem memória
(formulação alternativa)

Teorema:

"Uma fonte *X* de entropia *H(X)* produz símbolos à razão de um símbolo por cada T_s segundos. Consideremos um canal discreto sem memória de capacidade C_s (bits/símbolo!) e que seja usado uma vez em cada T_c segundos. Então, se

$$\frac{H(X)}{T_s} \leq \frac{C_s}{T_c},$$

existe uma técnica de codificação dos símbolos da fonte que permite que os símbolos transmitidos através do canal possam ser recuperados com uma probabilidade de erro arbitrariamente pequena.

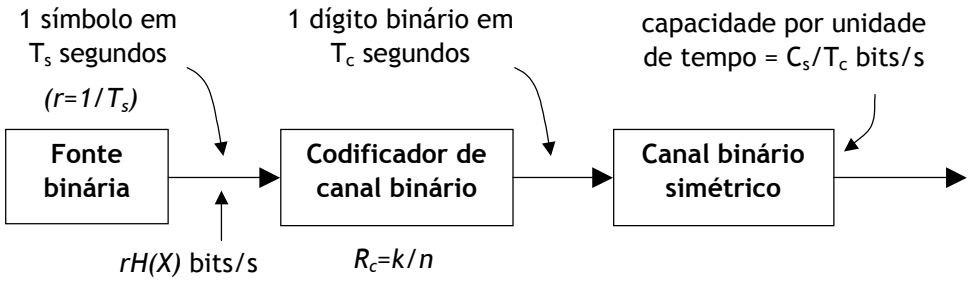
Se, pelo contrário, $\frac{H(X)}{T_s} > \frac{C_s}{T_c}$, então não é possível transmitir os símbolos e recuperá-los com uma probabilidade de erro arbitrariamente pequena."

Conversão de C_s (bits/símbolo) em C (bits/s):

$$C = \frac{C_s}{T_c}$$

Aplicação do "Teorema da Codificação do Canal" a um canal binário simétrico

Considere-se uma fonte binária a gerar símbolos equiprováveis de duração T_s segundos, ou seja, a produzi-los à taxa de $r = 1/T_s$ símbolos por segundo. Sendo os símbolos binários equiprováveis, a entropia da fonte vale $H(X) = 1$ bit/símbolo e a taxa de informação da fonte é $R = rH(X) = 1/T_s$ bits/s.



O codificador (de canal!) produz um símbolo binário em cada T_c segundos, o qual é transmitido através de um canal binário simétrico (BSC) – que, portanto, é usado uma vez em cada T_c segundos. Assim, a capacidade do canal por unidade de tempo é $C = C_s/T_c$ bits/s.

De acordo com o teorema da codificação do canal, terá de ser

$$\frac{H(X)}{T_s} \le \frac{C_s}{T_c} \Rightarrow \qquad \frac{1}{T_s} \le \frac{C_s}{T_c} \qquad \Rightarrow \qquad \frac{T_c}{T_s} \le C_s$$

Mas $\frac{T_c}{T_s} = R_c$ (taxa do código, ver a seguir), pelo que

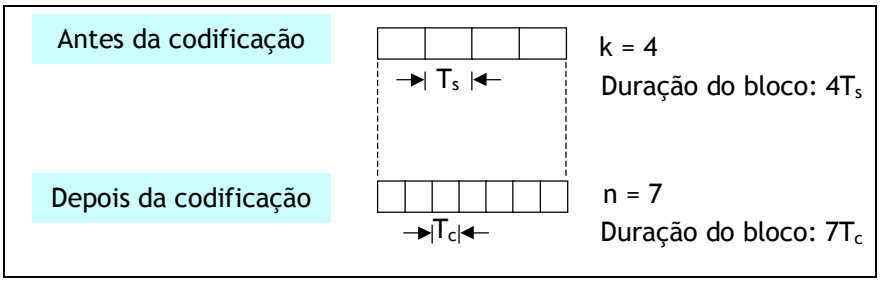
$$R_c \le C_s$$

Como a capacidade do canal BSC é $C_s = 1 - \Omega(p)$:

$$R_c \le 1 - \Omega(p) \qquad \text{(canal BSC)}$$

Aplicação do "Teorema da Codificação do Canal" a um canal binário simétrico (contin.)

Porque é que $\frac{T_c}{T_s} = R_c$?



À razão $\frac{k}{n} = R_c$ chama-se *taxa do código*.

Como a duração dos blocos é a mesma:

$$kT_s = nT_c$$

ou, neste caso,

$$4T_s = 7T_c \Rightarrow T_c/T_s = 4/7 = R_c$$

Em resumo, temos duas maneiras de exprimir o teorema da codificação de canal:

- Com a capacidade expressa em bits/símbolo, C_s: $R_c \le C_s$
- Com a capacidade expressa em bits/s, C: $R = rH(X) \le C$

bits/s ← $C = C_s/T_c$ ← bits/símbolo

1.4. Teorema da codificação de fonte

- Teorema da codificação de fonte
- Redundância
- códigos unicamente descodificáveis
- códigos sem prefixos
- desigualdade de Kraft-McMillan
- extensões

Codificação de fonte

Queremos transformar as mensagens de uma fonte num conjunto de símbolos de modo que seja "ocupado menos espaço" ou que a informação da fonte "demore menos tempo a ser transmitida". Queremos fazê-lo sem perdas: a operação inversa de descodificação ou descompressão deve dar origem exactamente às mesmas mensagens originais. Ora,

- A caracterização de uma fonte discreta é fácil se os símbolos forem independentes e equiprováveis (a entropia é máxima).
- Essa caracterização já é mais difícil se os símbolos não forem independentes (a probabilidade de um símbolo depende dos símbolos passados e a entropia é menor).
- Shannon provou que, se dividirmos a mensagem em blocos de letras ou palavras (no caso de texto), podemos calcular a entropia de cada bloco (= símbolo) pela mesma fórmula usada com símbolos independentes e assim aproximar-nos da entropia da fonte, considerando os blocos muito compridos.

Existe alguma relação entre a *entropia* e o *número médio de dígitos binários por símbolo* necessário para codificar uma mensagem? Sim, existe!

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

  No caso dos símbolos da fonte *não serem equiprováveis* o código óptimo tem em conta as diferentes probabilidades de ocorrência usando *palavras com comprimentos variáveis*.

- Convém agrupar os símbolos da fonte antes de codificar ou comprimir.

Representação de símbolos por dígitos binários: dois exemplos[1]

1. *Máquina de escrever com 48 teclas + CR + LF = 50 símbolos*

- Precisaríamos de log_2 50 = 5,62 dígitos binários/símbolo. A codificação faz-se atribuindo um número binário com 6 dígitos *a cada símbolo* (desperdiçando 2^6 - 50 = 14 números binários).

- Mas em 50 caracteres existem 50^3 = 125 000 grupos diferentes possíveis de 3 caracteres. Por outro lado 17 dígitos binários podem ser combinados de 131 072 maneiras diferentes (2^{17}).

  $\Rightarrow$ Se dividirmos o texto em blocos de 3 caracteres sucessivos poderemos atribuir *a cada bloco* um número binário de 17 dígitos.

Se tivessemos representado cada carácter por 6 dígitos binários precisávamos de 18 dígitos binários por cada grupo de 3 caracteres, em vez de 17.

Conclusão: agrupando os símbolos a compressão ou codificação é mais eficiente.

2. *Inglês (26 letras + espaço)*

- Codificação letra a letra, consideradas equiprováveis:

  São precisos 4,76 dígitos binários/carácter

- Codificação letra a letra tendo em conta as probabilidades relativas de ocorrência:

  São precisos 4,03 dígitos binários/carácter

- Codificação palavra a palavra tendo em conta a frequência relativa das palavras:

  São precisos 1,66 dígitos binários/carácter

Shannon calculou a entropia do inglês entre 0,6 e 0,13 bits/carácter.

[1] In J. R. Pierce, "An Introduction to Information Theory", 2ª edição, Dover, 1980.

Redundância

- A influência entre símbolos reduz a entropia.

  Exemplo: linguagem, textos escritos: cada letra depende da(s) precedente(s) ou até das palavras anteriores, devido a regras impostas pela linguagem.

  *A seguir à letra **Q** vem sempre **U**.*

- A influência mútua reduz a incerteza, logo, reduz a quantidade de informação.

  Trata-se de uma fonte *redundante*: significa qe há símbolos produzidos pla font qe não são absolutament essenciais para a transmssão da informação.

- A redundância de uma sequência de símbolos mede-se em função da redução de entropia ocorrida:

  Redundância: $E = 1 - \frac{H(Y|X)}{H(X)}$

  $H(Y|X)$ — supondo símbolos dependentes

  $H(X)$ — supondo símbolos independentes

  Num exemplo anterior em que $H(Y|X) = 0,9328$ bits/símbolo e $H(X) = 1,287$ bits/símbolo existe uma redundância na informação da fonte de E = 27,5%.

- Numa comunicação eficiente a redundância é indesejável (a mesma informação poderá ser transmitida usando menos símbolos).

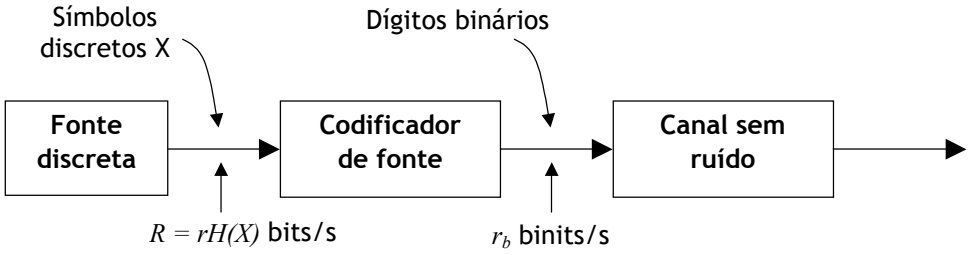
  MAS ... ajuda a resolver ambiguidades se a mensagem for recebiga com errus.

- A codificação para transmissão óptima pretende:
  - reduzir a redundância "ineficiente" → códigos de fonte (compressão)
  - acrescentar redundância "eficiente" → códigos de canal

Codificação de fonte

Consideremos uma fonte discreta de entropia *H(X)* a gerar símbolos à cadência de *r* símbolos/s. O ritmo de informação é, como se sabe, $R = rH(X)$ bits/s.

E se quisermos codificar estes símbolos através de dígitos binários?



Shannon mostrou que a informação proveniente de qualquer *fonte discreta sem memória* pode ser codificada como dígitos binários e transmitida através de um canal sem ruído à taxa binária de $r_b \geq R$ dígitos binários/s. Não é possível fazê-lo a uma taxa inferior ($r_b < R$).

Esta é uma manifestação do famoso "teorema da codificação de fonte", de Shannon, demonstrado mais adiante.

OBS: Aos dígitos binários vamos passar a chamar *binits*, para evitar confusões com a unidade de informação "bit". Só lhes deveremos chamar *bits* se não houver ambiguidades de interpretação.

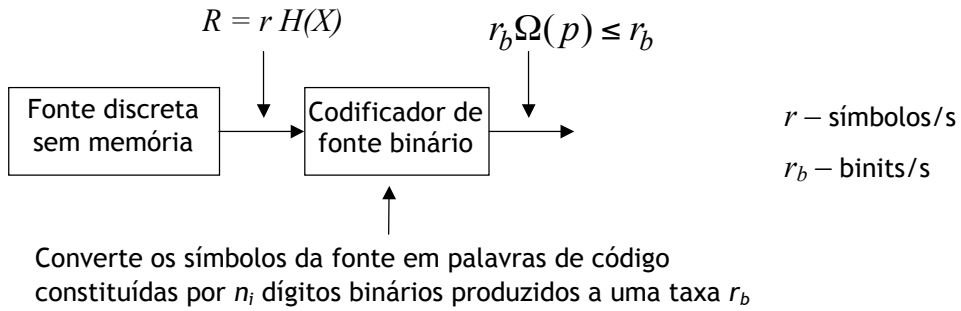
Teorema da codificação de fonte

- Fonte de M símbolos equiprováveis produzidos ao ritmo $r \rightarrow R = r \log M$

  Todos os símbolos transportam a mesma quantidade de informação.

- Fonte de símbolos com diferentes probabilidades $P_i \rightarrow R = rH(X) < r \log M$

- A transmissão eficiente requer um processo de **codificação** que tenha em conta a quantidade variável de informação por símbolo.



  - O codificador comporta-se como uma fonte binária com entropia $\Omega(p)$ e taxa de informação $r_b\Omega(p) \le r_b \log 2 = r_b$ binits/s.
  - O comprimento médio das palavras de código é igual a $\overline{N} = \sum_{i=1}^{M} P_i n_i$.

A codificação não gera informação adicional nem destrói a informação **SE** o código for decifrável de uma única maneira, sem ambiguidades *(códigos unicamente decifráveis ou descodificáveis)*. Nesse caso as taxas de informação à entrada e saída do codificador são iguais:

$$rH(X) = r_b\Omega(p) \le r_b \qquad \Rightarrow \qquad r_b \ge rH(X) \qquad \text{e} \qquad H(X) \le \frac{r_b}{r} = \overline{N}$$

(Teorema da codificação de fonte)

Codificação de fonte e ritmo de informação: um exemplo

Uma fonte emite r = 2000 símbolos/s seleccionados de um alfabeto de M = 4 elementos, com as probabilidades de ocorrência indicadas. Se quisermos codificar estes símbolos através de dígitos binários qual é o número mínimo de binits que podemos transmitir por unidade de tempo?

x_i	P_i	I_i
A	1/2	1
B	1/4	2
C	1/8	3
D	1/8	3

R.: A entropia desta fonte vale

$$H(X) = \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{8} \times 3 = 1,75 \text{ bits/símbolo}$$

(o máximo seria *log M = 2*)

A respectiva taxa de informação vale $R = 2000 \times 1,75 = 3500$ bits/s

Uma codificação de fonte apropriada permitirá transmitir a informação da fonte a uma taxa binária $r_b \geq 3500$ binits/s. Não é possível transmitir com menos dígitos binários por unidade de tempo.

Repare-se nas dimensões que nos aparecem neste exemplo:

[r] — símbolos/s [R] — bits/s [r_b] — binits/s

Teorema da codificação da fonte e desigualdade de Kraft

◊ Num código binário unicamente descodificável é condição necessária que os comprimentos n_i das palavras respeitem a desigualdade (ver Anexo 1)

$$K = \sum_{i=1}^{M} 2^{-n_i} \leq 1 \qquad \textit{(Desigualdade de Kraft)}$$

- Se o alfabeto do código tiver tamanho D a desigualdade de Kraft é

$$K = \sum_{i=1}^{M} D^{-n_i} \leq 1 \quad .$$

◊ O *teorema da codificação da fonte*, de Shannon, estabelece que, com códigos binários unicamente descodificáveis, o comprimento médio $\overline{N}$ é limitado por

$$H(X) \leq \overline{N} < H(X) + \varepsilon \qquad \varepsilon > 0, \text{ arbitrariamente pequeno}$$

- Com codificação óptima: $\overline{N} = H(X)$. Na prática: $\overline{N} > H(X)$.

- A eficiência da codificação mede-se calculando a razão

$$\frac{\text{entropia da fonte}}{\text{comprimento médio das palavras de código}}$$

- A eficiência é igual a $\dfrac{R}{r_b} = \dfrac{\not{r} H(X)}{\not{r} \overline{N}} = \dfrac{H(X)}{\overline{N}} \leq 1.$

- Codificação mais simples: código de comprimento fixo (exemplo: ASCII).

  Neste caso a desigualdade de Kraft toma a forma $K = M.2^{-\overline{N}} \leq 1$

  $$\Rightarrow \quad \overline{N} \geq \log_2 M \qquad \Rightarrow \qquad \text{eficiência} = \frac{H(X)}{\overline{N}} \leq \frac{H(X)}{\log M}$$

- Para aumentar a eficiência é preciso diminuir o comprimento médio $\overline{N}$, o que se consegue com os *códigos de comprimento variável*:

  Aos símbolos que ocorrem com mais frequência devem corresponder palavras de código mais curtas que as que correspondem aos símbolos mais raros.

Codificação binária de uma fonte discreta sem memória

EXEMPLO: fonte discreta sem memória produzindo 4 símbolos A, B, C e D com probabilidades 1/2, 1/4, 1/8 e 1/8, respectivamente.

Temos muitas possibilidades de codificação. Eis quatro possíveis:

x_i	P_i	Código I	Código II	Código III	Código IV
A	1/2	00	0	0	0
B	1/4	01	1	01	10
C	1/8	10	10	011	110
D	1/8	11	11	0111	111
	$\overline{N}$	2,0	1,25	1,875	1,75
	K	1,0	1,5	0,9375	1,0
		(comprimento fixo)		("comma code")	("tree code")

- H(X) = 1,75 bits/símbolo (exemplo anterior)
- Código I: Eficiência = $\frac{H(X)}{\overline{N}} = 88\%$
- Código II: $\overline{N} = 1{,}25 < H(X)$, $K = 2^{-1} + 2^{-1} + 2^{-2} + 2^{-2} = 1{,}5 > 1 \Rightarrow$ não é unicamente decifrável. (ex.:10011 pode ser BAABB ou CABB ou CAD, etc.)
- Código III: É unicamente decifrável (cada palavra começa por 0).

  Eficiência = $\frac{H(X)}{\overline{N}} = 93\%$
- Código IV: Nenhuma palavra de código aparece como o *prefixo* de outra palavra de código. É um código óptimo para esta fonte porque $\overline{N} = H(X)$ e $K = 1$.

  Eficiência = $\frac{H(X)}{\overline{N}} = 100\%$

  (Exemplo: 110010111 ↔ CABD)

Códigos unicamente descodificáveis

código unicamente descodificável	para cada sequência de fonte de comprimento finito a sequência de letras de código que lhe corresponde é diferente da sequência de letras de código correspondente a qualquer outra sequência de fonte.

Existe uma classe especial de códigos unicamente descodificáveis que satisfaz uma restrição chamada *condição de prefixação*. Chamar-lhes-emos *códigos sem prefixos*:

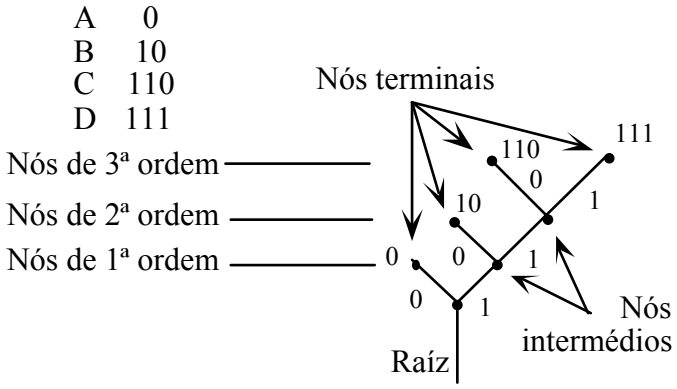
código sem prefixos (código instantâneo)	É um código no qual nenhuma palavra de código é o prefixo de qualquer outra palavra de código.

Seja $x_i = (x_{i,1}, \ldots, x_{i,ni})$ a palavra de código de ordem i. Qualquer sequência constituída por uma parte inicial de x_i (isto é, $(x_{i,1}, \ldots, x_{i,k})$, $k \leq n_i$) chama-se um *prefixo* de x_i.

Para descodificar uma sequência de palavras de código gerada por um código sem prefixos começa-se no princípio de cada palavra e descodifica-se uma a uma. Sabemos que chegámos ao fim de cada palavra porque essa palavra de código não é prefixo de qualquer outra (veja-se o anterior código IV, por exemplo).

Códigos sem prefixos

Representação gráfica através de "árvores"



- Num código sem prefixos as palavras de código terminam sempre nos *nós terminais*.

- Para descodificar, o receptor "sobe pela árvore", desde a raiz.

- Para que $\overline{N}$ — número médio de dígitos binários por símbolo de fonte — seja minimizado convém escolher as palavras de código de tal maneira que cada ramo que sai de um nó seja equiprovável (idealmente).

  Exemplo: Código de Shannon-Fano.

- Se cada ramo for equiprovável e o código tiver um alfabeto de D elementos, então

$P(a_i) = D^{-n_i}$ $\quad a_i$ – símbolo da fonte

n_i – comprimento da palavra de código correspondente

Exemplos de códigos de fonte sem prefixos:

- **Shannon-Fano**
- **Huffman.**

Teorema da codificação de fonte, de Shannon: demonstração

Teorema: *Dada uma fonte X com entropia H(X) e dado um alfabeto de código com D símbolos, é possível atribuir palavras de código às letras da fonte de tal maneira que a* ***condição de prefixação é satisfeita*** *e o comprimento médio das palavras de código,* $\overline{N}$*, satisfaz*

$$\overline{N} < \frac{H(X)}{\log D} + 1$$

Para qualquer conjunto ***unicamente descodificável*** *de palavras de código tem-se também*

$$\overline{N} \geq \frac{H(X)}{\log D}$$

Demonstração:

Mostremos que $\overline{N} \geq \frac{H(X)}{\log D}$, ou $H(X) - \overline{N} \log D \leq 0$.

Sejam $P(a_1)$, $P(a_2)$ ... $P(a_M)$ as probabilidades das letras da fonte e sejam n_1, ..., n_M os comprimentos das palavras de código respectivas:

$$H(X) - \overline{N} \log D = \sum_{i=1}^{M} P(a_i) \log \frac{1}{P(a_i)} - \sum_{i=1}^{M} P(a_i) n_i \log D =$$

$$= \sum_{i=1}^{M} P(a_i) \log \frac{D^{-n_i}}{P(a_i)}$$

Como $\log z \leq (z-1) \log e$ para z > 0, então

$$H(X) - \overline{N} \log D \leq \log e \sum_i P(a_i) [\frac{D^{-n_i}}{P(a_i)} - 1] = \log e \left[\overbrace{\sum_i D^{-n_i}}^{\leq 1} - \overbrace{\sum_i P(a_i)}^{1} \right]$$

$$\Rightarrow H(X) - \overline{N} \log D \leq 0$$

A igualdade verifica-se para

$$P(a_i) = D^{-n_i} \qquad 1 \leq i \leq M$$

(Continua)

Teorema da codificação de fonte

(Contin.)

Provemos agora que $\overline{N} < \frac{H(X)}{\log D} + 1$:

Como n_i tem de ser inteiro poderemos aproximar $P(a_i) = D^{-n_i}$ escolhendo n_i como o inteiro que satisfaz

$$D^{-n_i} \leq P(a_i) < D^{-n_i+1} \qquad 1 \leq i \leq M \qquad (*)$$

Somando em i:

$$\sum_i D^{-n_i} \leq \sum_i P(a_i) = 1 \qquad \text{(Kraft...)}$$

$\Rightarrow$ existe um código sem prefixos com estes comprimentos.

Quanto ao 2º membro de (*), tomemos logaritmos:

$$\log P(a_i) < (-n_i + 1)\log D$$

Multiplicando por $P(a_i)$ e somando em i:

$$\underbrace{\sum_i P(a_i)\log P(a_i)}_{-H(X)} < \sum_i P(a_i)\log D^{-n_i+1} = \left[\sum_i P(a_i)(-n_i+1)\right]\log D =$$

$$= \left[\underbrace{-\sum_i P(a_i)n_i}_{-\overline{N}} + \underbrace{\sum_i P(a_i)}_{1}\right]\log D$$

$$\Rightarrow -H(X) < (-\overline{N} + 1)\log D \qquad \Rightarrow \qquad -\frac{H(X)}{\log D} < -\overline{N} + 1$$

ou $\qquad \overline{N} < \frac{H(X)}{\log D} + 1 \qquad$ *c.q.d.*

Agrupamentos de símbolos de fontes ("extensões")

Seja *H(X)* a entropia de uma fonte discreta *X* que gera símbolos independentes retirados de um alfabeto com *M* símbolos. Agrupando os símbolos de *X* em grupos de *L* símbolos obtemos um conjunto aumentado, X^L, de M^L símbolos chamado "extensão de ordem *L*" de *X*. A entropia da nova fonte X^L está relacionada com a entropia de *X* através de

$$H(X^L) = LH(X)$$

- *Exemplo 9.2 de Haykin, 4ª edição (pág. 573):*

As probabilidades dos símbolos de uma fonte ternária (*M=3*) são

$$P_0 = 1/4$$
$$P_1 = 1/4$$
$$P_2 = 1/2$$

A entropia desta fonte vale $H(1/4,1/4,1/2) = 1{,}5$ bits/símbolo. Uma extensão de segunda ordem produz nove (M^2) símbolos com as seguintes probabilidades:

Símbolos de X^2	σ_0	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7	σ_8
Sequências correspondentes de símbolos de X	x_0x_0	x_0x_1	s_0s_2	x_1x_0	x_1x_1	x_1x_2	x_2x_0	x_2x_1	x_2x_2
$p(\sigma_i)$, $i = 0,1,\ldots 8$	1/16	1/16	1/8	1/16	1/16	1/8	1/8	1/8	1/4

A entropia da fonte aumentada vale, portanto,

$$H(X^2) = -\sum_{i=0}^{8} p(\sigma_i)\log p(\sigma_i) = 3,$$

que é precisamente igual ao dobro de *H(X)*, como se esperava.

Teorema da codificação de fonte

Aplicação a "fontes aumentadas" através de extensões ou agrupamentos de símbolos

Admitamos que os símbolos de X^L são codificados usando um código unicamente descodificável. Se $\overline{N_L}$ for o comprimento médio, por símbolo de X^L, das respectivas palavras de código, $\overline{N} = \overline{N_L}/L$ representa o comprimento médio por símbolo de *X*. O *Teorema da Codificação de Fonte* indica que

$$\frac{H(X^L)}{\log_2 D} \leq \overline{N_L} < \frac{H(X^L)}{\log_2 D} + 1$$

Como $H(X^L) = LH(X)$, poderemos escrever

$$\frac{LH(X)}{\log_2 D} \leq \overline{N_L} < \frac{LH(X)}{\log_2 D} + 1 \Rightarrow \frac{H(X)}{\log_2 D} \leq \overline{N_L}/L < \frac{H(X)}{\log_2 D} + 1/L$$

isto é,

$$\frac{H(X)}{\log_2 D} \leq \overline{N} < \frac{H(X)}{\log_2 D} + 1/L$$

Se não se tivesse feito um agrupamento antes da codificação (ou seja, se $L = 1$), teríamos

$$\frac{H(X)}{\log_2 D} \leq \overline{N} < \frac{H(X)}{\log_2 D} + 1,$$

que representa um intervalo maior para o comprimento médio das palavras de código.

Caso de fontes binárias:

Passamos a ter expressões simplificadas:

$$H(X^L) \leq \overline{N_L} < H(X^L) + 1$$

$$H(X) \leq \overline{N} < H(X) + 1/L$$

Codificação de fonte: considerações diversas

- Embora não possamos controlar a estatística da fonte de informação, a codificação de fonte deve obedecer ao seguinte (como nos códigos de Morse, Huffman, etc.):

  > *Aos símbolos mais frequentes devem corresponder palavras de código curtas; aos símbolos menos frequentes devem corresponder palavras compridas.*

- Entropia de fonte binária = 1 bit/símbolo se os símbolos forem equiprováveis.

- Se os símbolos *não forem equiprováveis* a informação média por símbolo é inferior a um e a fonte apresenta *redundância*.

- A codificação da fonte reduz essa redundância por forma a aumentar a eficiência da comunicação.

- A codificação faz-se agrupando a sequência de dígitos binários da fonte em blocos de n símbolos, passando a formar um novo conjunto de 2^n símbolos.

- A probabilidade de cada um dos 2^n novos símbolos é calculada e atribui-se a palavra de código mais curta ao novo símbolo mais provável, e assim por diante: palavras mais compridas correspondem a símbolos menos frequentes.

  Ao conjunto de 2^n novos símbolos chama-se *extensão de ordem n* da fonte binária.

Exemplo de codificação de fonte binária com extensões de 2ª e 3ª ordem

i	$S_i^{(1)}$	$P\left[S_i^{(1)}\right]$	Palavras de código
1	A	0,9	0
2	B	0,1	1

Fonte original:

$H(X) = 0,469$ bits/símbolo

$\overline{N} = \frac{\overline{N}_1}{1} = 1,0$ binit

i	$S_i^{(2)}$	$P\left[S_i^{(2)}\right]$	Palavras de código
1	AA	0,81	0
2	AB	0,09	10
3	BA	0,09	110
4	BB	0,01	111

Extensão de 2ª ordem

$\overline{N} = \frac{\overline{N}_2}{2} = 0,645$ binits

i	$S_i^{(3)}$	$P\left[S_i^{(3)}\right]$	Palavras de código
1	AAA	0,729	0
2	AAB	0,081	100
3	ABA	0,081	101
4	BAA	0,081	110
5	ABB	0,009	11100
6	BAB	0,009	11101
7	BBA	0,009	11110
8	BBB	0,001	11111

Extensão de 3ª ordem

$\overline{N} = \frac{\overline{N}_3}{3} = 0,5333$ binits

$\overline{N}_n = \sum_{i=1}^{2^n} N_i P\left[S_i^{(n)}\right]$ — comprimento médio das palavras de código para a extensão de ordem n.

$\overline{N} = \frac{\overline{N}_n}{n}$ — número médio de símbolos de código por símbolo de fonte.

$n \uparrow \Rightarrow \overline{N} = \frac{\overline{N}_n}{n} \downarrow$ $\Rightarrow$ menor taxa de símbolos

(Continua)

Exemplo de codificação de fonte binária com extensões de 2ª e 3ª ordem

Exemplo com modulação PSK binária:

- Potência recebida = $10^{-5} W$
- Taxa da fonte = 1000 símbolos/s$\Rightarrow$ Energia de bit: $E_b = 10^{-8} J$
- $N_0 = 10^{-9} W/Hz$ (densidade espectral de potência do ruído branco)

Se a potência de emissão for fixa a energia de símbolo pode ser aumentada usando uma extensão porque a taxa de símbolos passa a ser menor.

Em PSK binário a probabilidade de erro sem codificação é

$$P_B = Q\left(\sqrt{\frac{2E_b}{N_0}}\right).$$

Substituindo valores e consultando uma tabela da função Q, teremos:

$$P_B = Q\left(\sqrt{\frac{2\times 10^{-8}}{10^{-9}}}\right) = Q\left(\sqrt{20}\right) = 3{,}9.10^{-6}$$

Usemos agora extensões:

- Extensão de 2ª ordem $\Rightarrow$ 645 símbolos/s $\Rightarrow E_b = 1{,}55.10^{-8} J \Rightarrow P_B = 1{,}3.10^{-8}$
- Extensão de 3ª ordem $\Rightarrow$ 533,3 símbolos/s $\Rightarrow P_B = 4{,}5.10^{-10}$

Como se vê…

1.5. Códigos de fonte para compressão de dados

- técnicas de compressão sem perdas
- códigos baseados em modelos estatísticos e em dicionários
    - código de Shannon-Fano
    - código de Huffman
    - código aritmético
    - códigos de Lempel-Ziv (LZ77 e LZ78)

Compressão de dados

Técnicas de compressão sem perdas

Modelos estatísticos

Técnicas baseadas em dicionários

Exemplos de códigos

- Shannon-Fano
- Huffman
- Codificação aritmética

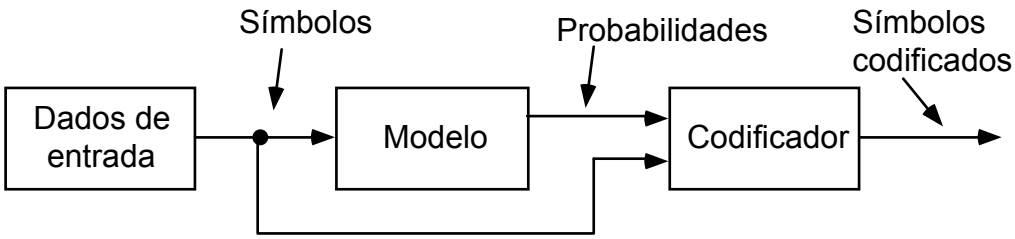
- Lempel-Ziv (LZ 77)
- Lempel-Ziv (LZ 78)
- Lempel-Ziv-Welch (LZW)

Compressão de dados

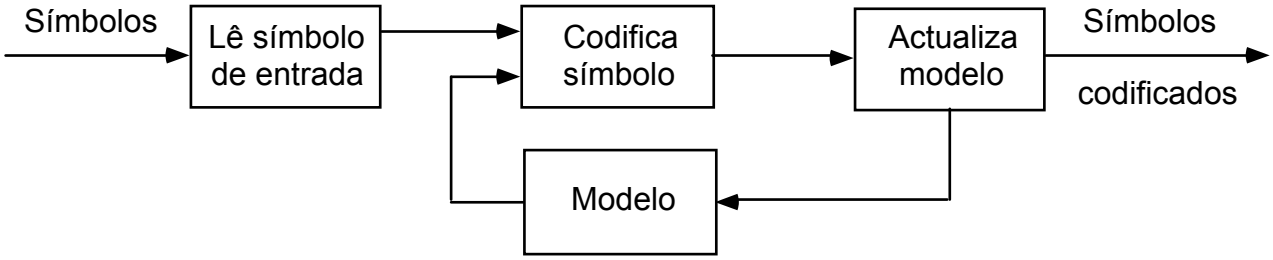
Compressão = Modelização + Codificação

Se o modelo não fornecer probabilidades correctas o codificador comprime pouco, por muito eficiente que seja.

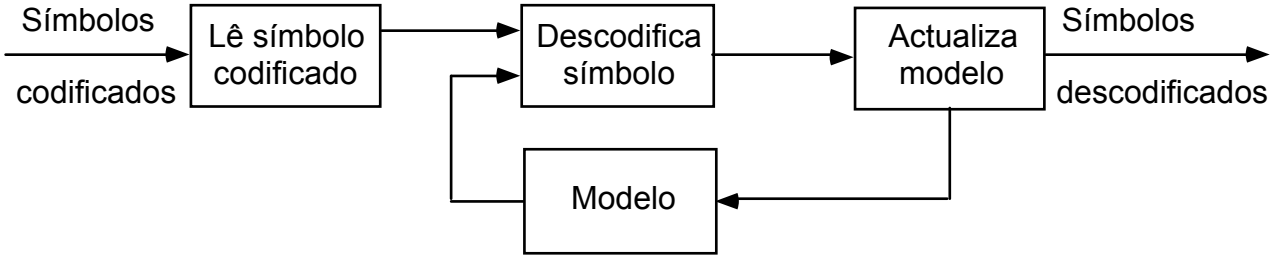
Modelo estatístico



Compressão adaptativa genérica



Descompressão adaptativa genérica



Porquê métodos de compressão adaptativos?

- Huffman (estático): as estatísticas da fonte têm de ser passadas ao programa de expansão (descodificação).

$\Downarrow$

Essa informação é um "overhead", que tanto pode ser pequeno como muitíssimo grande (depende do rigor do modelo).

- Num modelo adaptativo não se passa informação estatística ao descodificador

$\Downarrow$

não há "overhead".

Mas… de início o codificador está "às escuras", isto é, não conhece as estatísticas da fonte porque ainda não começou a determiná-las.

$\Downarrow$

Isto quer dizer que:

de início comprime mal os dados mas "vai melhorando com o tempo!"

Compressão de dados

Marcos históricos

1952: D. A. Huffman, “A method for the construction of minimum redundancy codes”, Proceedings of the IRE, Setembro de 1952.

1977: J. Ziv e A. Lempel, “A universal algorithm for sequential data compression”, IEEE Transactions on Information Theory, Maio de 1977.

1978: J. Ziv e A. Lempel, “Compression of individual sequences via variable rate coding”, IEEE Transactions on Information Theory, Setembro de 1978.

1984: T. Welch, “A technique for high-performance data compression”, IEEE Computer, Junho de 1984.

O artigo (comprimido) de David Huffman

A Method for the Construction of Minimum-Redundancy Codes*

DAVID A. HUFFMAN†, ASSOCIATE, IRE

Summary—An optimum method of coding an ensemble of messages consisting of a finite number of members is developed. A minimum-redundancy code is one constructed in such a way that the average number of coding digits per message is minimized.

Introduction

ONE IMPORTANT METHOD of transmitting messages is to transmit in their place sequences of symbols. If there are more messages which might be sent than there are kinds of symbols available, then some of the messages must use more than one symbol. If it is assumed that each symbol requires the same time for transmission, then the time for transmission (length) of a message is directly proportional to the number of symbols associated with it. In this paper, the symbol or sequence of symbols associated with a given message will be called the "message code." The entire number of messages which might be transmitted will be called the "message ensemble." The mutual agreement between the transmitter and the receiver about the meaning of the code for each message of the ensemble will be called the "ensemble code."

Probably the most familiar ensemble code was stated in the phrase "one if by land and two if by sea." In this case, the message ensemble consisted of the two individual messages "by land" and "by sea", and the message codes were "one" and "two."

In order to formalize the requirements of an ensemble code, the coding symbols will be represented by numbers. Thus, if there are D different types of symbols to be used in coding, they will be represented by the digits $0, 1, 2, \cdots, (D-1)$. For example, a ternary code will be constructed using the three digits 0, 1, and 2 as coding symbols.

The number of messages in the ensemble will be called N. Let $P(i)$ be the probability of the ith message. Then

$$\sum_{i=1}^{N} P(i) = 1. \tag{1}$$

The length of a message, $L(i)$, is the number of coding digits assigned to it. Therefore, the average message length is

$$L_{av} = \sum_{i=1}^{N} P(i)L(i). \tag{2}$$

The term "redundancy" has been defined by Shannon[1] as a property of codes. A "minimum-redundancy code" will be defined here as an ensemble code which, for a message ensemble consisting of a finite number of members, N, and for a given number of coding digits, D, yields the lowest possible average message length. In order to avoid the use of the lengthy term "minimum-redundancy," this term will be replaced here by "optimum." It will be understood then that, in this paper, "optimum code" means "minimum-redundancy code."

The following basic restrictions will be imposed on an ensemble code:

(a) No two messages will consist of identical arrangements of coding digits.

(b) The message codes will be constructed in such a way that no additional indication is necessary to specify where a message code begins and ends once the starting point of a sequence of messages is known.

Restriction (b) necessitates that no message be coded in such a way that its code appears, digit for digit, as the first part of any message code of greater length. Thus, 01, 102, 111, and 202 are valid message codes for an ensemble of four members. For instance, a sequence of these messages 111102202010111102 can be broken up into the individual messages 111-102-202-01-01-111-102. All the receiver need know is the ensemble code. However, if the ensemble has individual message codes including 11, 111, 102, and 02, then when a message sequence starts with the digits 11, it is not immediately certain whether the message 11 has been received or whether it is only the first two digits of the message 111. Moreover, even if the sequence turns out to be 11102, it is still not certain whether 111-02 or 11-102 was transmitted. In this example, change of one of the two message codes 111 or 11 is indicated.

C. E. Shannon[1] and R. M. Fano[2] have developed ensemble coding procedures for the purpose of proving that the average number of binary digits required per message approaches from above the average amount of information per message. Their coding procedures are not optimum, but approach the optimum behavior when N approaches infinity. Some work has been done by Kraft[3] toward deriving a coding method which gives an average code length as close as possible to the ideal when the ensemble contains a finite number of members. However, up to the present time, no definite procedure has been suggested for the construction of such a code

* Decimal classification: R531.1. Original manuscript received by the Institute, December 6, 1951.

† Massachusetts Institute of Technology, Cambridge, Mass.

[1] C. E. Shannon, "A mathematical theory of communication," Bell Sys. Tech. Jour., vol. 27, pp. 398–403; July, 1948.

[2] R. M. Fano, "The Transmission of Information," Technical Report No. 65, Research Laboratory of Electronics, M.I.T., Cambridge, Mass.; 1949.

[3] L. G. Kraft, "A Device for Quantizing, Grouping, and Coding Amplitude-modulated Pulses," Electrical Engineering Thesis, M.I.T., Cambridge, Mass.; 1949.

Reprinted from *Proc. IRE*, vol. 40, pp. 1098–1101, Sept. 1952.

to the knowledge of the author. It is the purpose of this paper to derive such a procedure.

Derived Coding Requirements

For an optimum code, the length of a given message code can never be less than the length of a more probable message code. If this requirement were not met, then a reduction in average message length could be obtained by interchanging the codes for the two messages in question in such a way that the shorter code becomes associated with the more probable message. Also, if there are several messages with the same probability, then it is possible that the codes for these messages may differ in length. However, the codes for these messages may be interchanged in any way without affecting the average code length for the message ensemble. Therefore, it may be assumed that the messages in the ensemble have been ordered in a fashion such that

$$P(1) \geq P(2) \geq \cdots \geq P(N-1) \geq P(N) \tag{3}$$

and that, in addition, for an optimum code, the condition

$$L(1) \leq L(2) \leq \cdots \leq L(N-1) \leq L(N) \tag{4}$$

holds. This requirement is assumed to be satisfied throughout the following discussion.

It might be imagined that an ensemble code could assign q more digits to the Nth message than to the $(N-1)$st message. However, the first $L(N-1)$ digits of the Nth message must not be used as the code for any other message. Thus the additional q digits would serve no useful purpose and would unnecessarily increase L_{av}. Therefore, for an optimum code it is necessary that $L(N)$ be equal to $L(N-1)$.

The kth prefix of a message code will be defined as the first k digits of that message code. Basic restriction (b) could then be restated as: No message shall be coded in such a way that its code is a prefix of any other message, or that any of its prefixes are used elsewhere as a message code.

Imagine an optimum code in which no two of the messages coded with length $L(N)$ have identical prefixes of order $L(N)-1$. Since an optimum code has been assumed, then none of these messages of length $L(N)$ can have codes or prefixes of any order which correspond to other codes. It would then be possible to drop the last digit of all of this group of messages and thereby reduce the value of L_{av}. Therefore, in an optimum code, it is necessary that at least two (and no more than D) of the codes with length $L(N)$ have identical prefixes of order $L(N)-1$.

One additional requirement can be made for an optimum code. Assume that there exists a combination of the D different types of coding digits which is less than $L(N)$ digits in length and which is not used as a message code or which is not a prefix of a message code. Then this combination of digits could be used to replace the code for the Nth message with a consequent reduction of L_{av}. Therefore, all possible sequences of $L(N)-1$ digits must be used either as message codes, or must have one of their prefixes used as message codes.

The derived restrictions for an optimum code are summarized in condensed form below and considered in addition to restrictions (a) and (b) given in the first part of this paper:

(c) $L(1) \leq L(2) \leq \cdots \leq L(N-1) = L(N)$. (5)

(d) At least two and not more than D of the messages with code length $L(N)$ have codes which are alike except for their final digits.

(e) Each possible sequence of $L(N)-1$ digits must be used either as a message code or must have one of its prefixes used as a message code.

Optimum Binary Code

For ease of development of the optimum coding procedure, let us now restrict ourselves to the problem of binary coding. Later this procedure will be extended to the general case of D digits.

Restriction (c) makes it necessary that the two least probable messages have codes of equal length. Restriction (d) places the requirement that, for D equal to two, there be only two of the messages with coded length $L(N)$ which are identical except for their last digits. The final digits of these two codes will be one of the two binary digits, 0 and 1. It will be necessary to assign these two message codes to the Nth and the $(N-1)$st messages since at this point it is not known whether or not other codes of length $L(N)$ exist. Once this has been done, these two messages are equivalent to a single composite message. Its code (as yet undetermined) will be the common prefixes of order $L(N)-1$ of these two messages. Its probability will be the sum of the probabilities of the two messages from which it was created. The ensemble containing this composite message in the place of its two component messages will be called the first auxiliary message ensemble.

This newly created ensemble contains one less message than the original. Its members should be rearranged if necessary so that the messages are again ordered according to their probabilities. It may be considered exactly as the original ensemble was. The codes for each of the two least probable messages in this new ensemble are required to be identical except in their final digits; 0 and 1 are assigned as these digits, one for each of the two messages. Each new auxiliary ensemble contains one less message than the preceding ensemble. Each auxiliary ensemble represents the original ensemble with full use made of the accumulated necessary coding requirements.

The procedure is applied again and again until the number of members in the most recently formed auxiliary message ensemble is reduced to two. One of each of the binary digits is assigned to each of these two composite messages. These messages are then combined to form a single composite message with probability unity, and the coding is complete.

TABLE I

Optimum Binary Coding Procedure

Original Message Ensemble	Message Probabilities: Auxiliary Message Ensembles											
	1	2	3	4	5	6	7	8	9	10	11	12
0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.24	0.36	0.40	0.60	1.00
0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.20	0.20	0.24	0.36	0.40	
0.10	0.10	0.10	0.10	0.10	0.14	0.14	0.18	0.18	0.20	0.24		
0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.14	0.18	0.20			
0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.14				
				0.10	0.10	0.10	0.10					
		0.08	0.08	0.08	0.08	0.08						
0.06	0.06	0.06	0.06	0.06	0.06							
0.04	0.04	0.04	0.04	0.04								
0.04	0.04	0.04	0.04									
0.04	0.04	0.04										
0.03	0.04											
0.01												

Now let us examine Table I. The left-hand column contains the ordered message probabilities of the ensemble to be coded. N is equal to 13. Since each combination of two messages (indicated by a bracket) is accompanied by the assigning of a new digit to each, then the total number of digits which should be assigned to each original message is the same as the number of combinations indicated for that message. For example, the message marked *, or a composite of which it is a part, is combined with others five times, and therefore should be assigned a code length of five digits.

When there is no alternative in choosing the two least probable messages, then it is clear that the requirements, established as necessary, are also sufficient for deriving an optimum code. There may arise situations in which a choice may be made between two or more groupings of least likely messages. Such a case arises, for example, in the fourth auxiliary ensemble of Table I. Either of the messages of probability 0.08 could have been combined with that of probability 0.06. However, it is possible to rearrange codes in any manner among equally likely messages without affecting the average code length, and so a choice of either of the alternatives could have been made. Therefore, the procedure given is always sufficient to establish an optimum binary code.

The lengths of all the encoded messages derived from Table I are given in Table II.

Having now determined proper lengths of code for each message, the problem of specifying the actual digits remains. Many alternatives exist. Since the combining of messages into their composites is similar to the successive confluences of trickles, rivulets, brooks, and creeks into a final large river, the procedure thus far described might be considered analogous to the placing of signs by a water-borne insect at each of these junctions as he journeys downstream. It should be remembered that the code which we desire is that one which the insect must remember in order to work his way upstream. Since the placing of the signs need not follow the same rule, such as "zero-right-returning," at each junction, it can be seen that there are at least 2^{13} different ways of assigning code digits for our example.

TABLE II

Results of Optimum Binary Coding Procedure

i	$P(i)$	$L(i)$	$P(i)L(i)$	Code
1	0.20	2	0.40	10
2	0.18	3	0.54	000
3	0.10	3	0.30	011
4	0.10	3	0.30	110
5	0.10	3	0.30	111
6	0.06	4	0.24	0101
7	0.06	5	0.30	00100
8	0.04	5	0.20	00101
9	0.04	5	0.20	01000
10	0.04	5	0.20	01001
11	0.04	5	0.20	00110
12	0.03	6	0.18	001110
13	0.01	6	0.06	001111
			$L_{av} = 3.42$	

The code in Table II was obtained by using the digit 0 for the upper message and the digit 1 for the lower message of any bracket. It is important to note in Table I that coding restriction (e) is automatically met as long as two messages (and not one) are placed in each bracket.

Generalization of the Method

Optimum coding of an ensemble of messages using three or more types of digits is similar to the binary coding procedure. A table of auxiliary message ensembles similar to Table I will be used. Brackets indicating messages combined to form composite messages will be used in the same way as was done in Table I. However, in order to satisfy restriction (e), it will be required that all these brackets, with the possible exception of one combining the least probable messages of the original ensemble, always combine a number of messages equal to D.

It will be noted that the terminating auxiliary ensemble always has one unity probability message. Each preceding ensemble is increased in number by $D-1$ until the first auxiliary ensemble is reached. Therefore, if N_1 is the number of messages in the first auxiliary ensemble, then $(N_1-1)/(D-1)$ must be an integer. However $N_1 = N - n_0 + 1$, where n_0 is the number of the least probable messages combined in a bracket in the original ensemble. Therefore, n_0 (which, of course, is at least two and no more than D) must be of such a value that $(N-n_0)/(D-1)$ is an integer.

In Table III an example is considered using an ensemble of eight messages which is to be coded with four digits; n_0 is found to be 2. The code listed in the table is obtained by assigning the four digits 0, 1, 2, and 3, in order, to each of the brackets.

TABLE III

Optimum Coding Procedure for $D=4$

Original Message Ensemble	Auxiliary Ensembles				$L(i)$	Code
				1.00		
0.22	0.22	0.22	0.40		1	1
0.20	0.20	0.20	0.22		1	2
0.18	0.18	0.18	0.20		1	3
0.15	0.15	0.15			2	00
0.10	0.10				2	01
0.08	0.08				2	02
	0.07					
0.05					3	030
0.02					3	031

Acknowledgments

The author is indebted to Dr. W. K. Linvill and Dr. R. M. Fano, both of the Massachusetts Institute of Technology, for their helpful criticism of this paper.

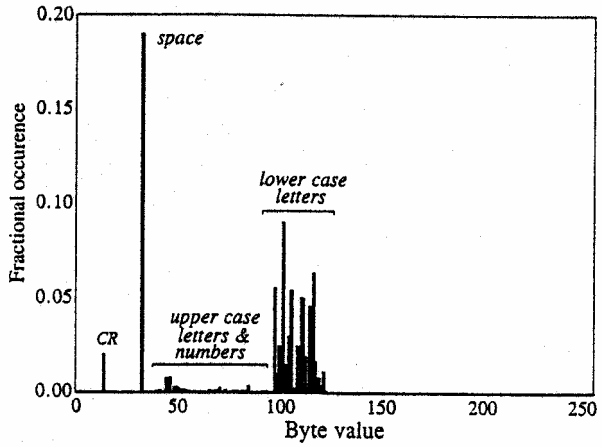
Um exemplo simples de compressão

Um texto é codificado em ASCII (8 bits/letra, de acordo com a tabela seguinte).

TABLE 27-2
ASCII codes. This is a long established standard for allowing letters and numbers to be represented in digital form. Each printable character is assigned a number between 32 and 127, while the numbers between 0 and 31 are used for various control actions. Even though only 128 codes are defined, ASCII characters are usually stored as a full byte (8 bits). The undefined values (128 to 255) are often used for Greek letters, math symbols, and various geometric patterns; however, this is not standardized. Many of the control characters (0 to 31) are based on older communications networks, and are not applicable to computer technology.

0	null	32	space	64	@	96	`
1	start heading	33	!	65	A	97	a
2	start of text	34	"	66	B	98	b
3	end of text	35	#	67	C	99	c
4	end of xmit	36	$	68	D	100	d
5	enquiry	37	%	69	E	101	e
6	acknowledge	38	&	70	F	102	f
7	bell, beep	39	'	71	G	103	g
8	backspace	40	(	72	H	104	h
9	horz. tab	41	)	73	I	105	i
10	line feed	42	*	74	J	106	j
11	vert. tab, home	43	+	75	K	107	k
12	form feed, cls	44	,	76	L	108	l
13	carriage return	45	-	77	M	109	m
14	shift out	46	.	78	N	110	n
15	shift in	47	/	79	O	111	o
16	data line esc	48	0	80	P	112	p
17	device control 1	49	1	81	Q	113	q
18	device control 2	50	2	82	R	114	r
19	device control 3	51	3	83	S	115	s
20	device control 4	52	4	84	T	116	t
21	negative ack.	53	5	85	U	117	r
22	synch. idle	54	6	86	V	118	v
23	end xmit block	55	7	87	W	119	w
24	cancel	56	8	88	X	120	x
25	end of medium	57	9	89	Y	121	y
26	substitute	58	:	90	Z	122	z
27	escape	59	;	91	[	123	{
28	file separator	60	<	92	\	124	|
29	group separator	61	=	93	]	125	}
30	record separator	62	>	94	^	126	~
31	unit separator	63	?	95	_	127	del

Suponhamos que a frequência relativa com que os caracteres ocorrem é representada pelo seguinte histograma. Os caracteres mais frequentes são as minúsculas, o espaço e o retorno ao início de linha ("carriage return", CR).



Verificou-se que 96% do texto é constituído por 31 caracteres (letras minúsculas, espaços, vírgulas, etc.).

⇒ Cada um desses 31 caracteres pode ser representado por 5 bits. A 32ª palavra de 5 bits (11111) pode ser uma "flag" indicando que a seguir vem um carácter que não é um dos 31 anteriores.

⇒ Cada um desses outros caracteres é representado por 5 + 8 = 13 bits.

⇒ Comprimento médio das palavras do código: $\overline{N} = 0,96 \times 5 + 0,04 \times 13 = 5,32$ bits (em vez dos 8 originais).

Código de Shannon-Fano

No Código de Shannon-Fano as palavras a codificar devem ser ordenadas por ordem decrescente de probabilidades. Suponhamos, por exemplo, que temos as palavras $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, $\boldsymbol{x}_4$, $\boldsymbol{x}_5$ e $\boldsymbol{x}_6$, que ocorrem com as seguintes probabilidades:

$P(x_1) = 0{,}1 \qquad P(x_2) = 0{,}2 \qquad P(x_3) = 0{,}2$

$P(x_4) = 0{,}03 \qquad P(x_5) = 0{,}07 \qquad P(x_6) = 0{,}4$

Ordenando os símbolos e dividindo-os sucessivamente em dois grupos de probabilidades aproximadas, obteremos o seguinte:

Símbolo	Probabilidade de ocorrência	Construção das palavras de código				Palavras de código
$P(x_6)$	0,4	0	0			00
$P(x_2)$	0,2	0	1			01
$P(x_3)$	0,2	1	0			10
$P(x_1)$	0,1	1	1	0		110
$P(x_5)$	0,07	1	1	1	0	1110
$P(x_4)$	0,03	1	1	1	1	1111

Entropia da fonte:

$$H(X) = -\sum_{i=1}^{6} P_i \log P_i =$$

$$= -(0{,}4\log_2 0{,}4 + 0{,}2\log_2 0{,}2 + +0{,}2\log_2 0{,}2 + 0{,}1\log_2 0{,}1 + 0{,}07\log_2 0{,}07 + 0{,}03\log_2 0{,}03) =$$

$$= 2{,}21 \text{ bits/símbolo}$$

Comprimento médio das palavras de código:

$$\overline{N} = 2 \times 0{,}4 + 2 \times 0{,}2 + 2 \times 0{,}2 + 3 \times 0{,}1 + 4 \times 0{,}07 + 4 \times 0{,}03 =$$

$$= 2{,}3 \text{ dígitos binários / símbolo}$$

$$\textbf{Eficiência do código} = \frac{\text{Entropia da fonte}}{\text{comprimento médio das palavras de código}} = \frac{H(X)}{\overline{N}} = \frac{2{,}21}{2{,}3} = 96\%$$

Código de Huffman binário

Este código de fonte é óptimo no sentido de que o número médio de dígitos binários necessário para representar o símbolo da fonte é mínimo.

Vamos usar o código de Huffman com uma fonte que produz oito símbolos:

$P(x_1) = 0{,}50 \quad P(x_2) = 0{,}02 \quad P(x_3) = 0{,}03 \quad P(x_4) = 0{,}04$

$P(x_5) = 0{,}10 \quad P(x_6) = 0{,}04 \quad P(x_7) = 0{,}12 \quad P(x_8) = 0{,}15$

Passos a tomar para construir a árvore do código (ver exemplo)**:**

1. Ordenar as mensagens por ordem decrescente de probabilidades.

   $x_1 \quad x_8 \quad x_7 \quad x_5 \quad x_4 \quad x_6 \quad x_3 \quad x_2$

2. Agrupar as duas mensagens com as probabilidades mais baixas e somar estas probabilidades.

   *Agrupamos x_2 e x_3 no nó **a**, cuja soma de probabilidades dá 0,05. De um conjunto de oito probabilidades passámos a ter um conjunto de 6+1=7 probabilidades.*

3. Repetir o passo anterior para o novo conjunto de probabilidades, até chegar à raiz da árvore.

   *3.1 Agrupamos x_4 e x_6 no nó b, cuja soma de probabilidades dá 0,08. De um conjunto de sete probabilidades passámos a ter um conjunto de seis probabilidades: 0,50, 0,15, 0,12, 0,10, 0,08 (nó **b**) e 0,05 (nó **a**).*

   *3.2 Agora agrupamos os nós **a** e **b**, produzindo o nó **c**. Depois agrupamos x_5 e x_7, produzindo o nó **d**, e assim por diante, gerando os nós **e** e **f** e chegando finalmente à raiz.*

4. Da copa para a raiz arbitrar o bit 1 (ou 0) para os ramos que partem do nó de menor probabilidade e o bit 0 (ou 1) para os que partem do outro nó.

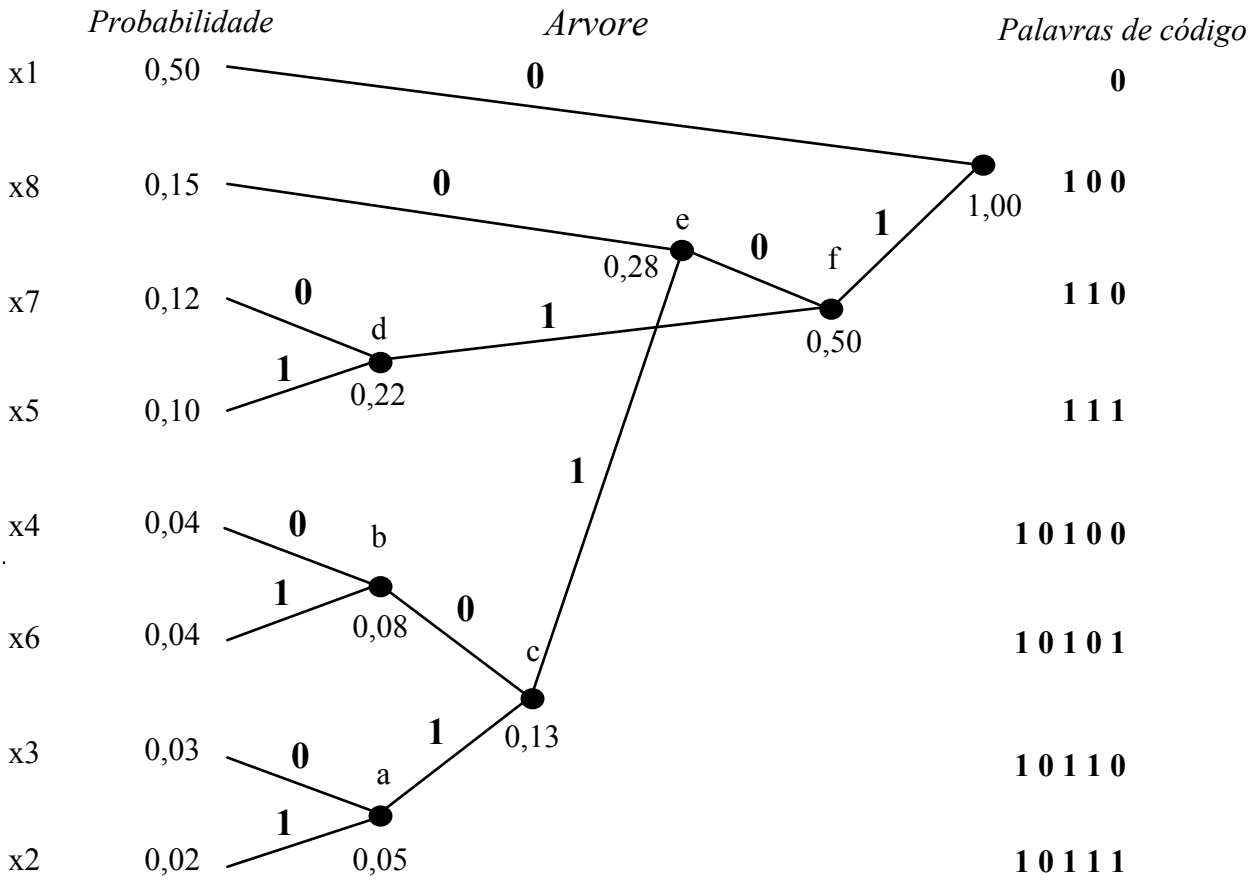
   *Por exemplo, ao ramo que vai do nó **a** para o nó **c** atribuamos-lhe o bit 1.*

5. Determinar as palavras de código percorrendo a árvore da raiz para a copa.

   Por exemplo, ao símbolo x_7 corresponde a palavra de código binária 110.

A entropia da fonte vale $H(X) = 2{,}21$ bits/símbolo e $\overline{N} = 2{,}26$. Se tomássemos palavras de código de igual comprimento necessitaríamos de 3 binits/símbolo. Assim bastou tomar 2,26 binits/símbolo $\Rightarrow$ Eficiência $= \frac{2{,}21}{2{,}26} = 97{,}3\%$.

A árvore do exemplo de código de Huffman binário



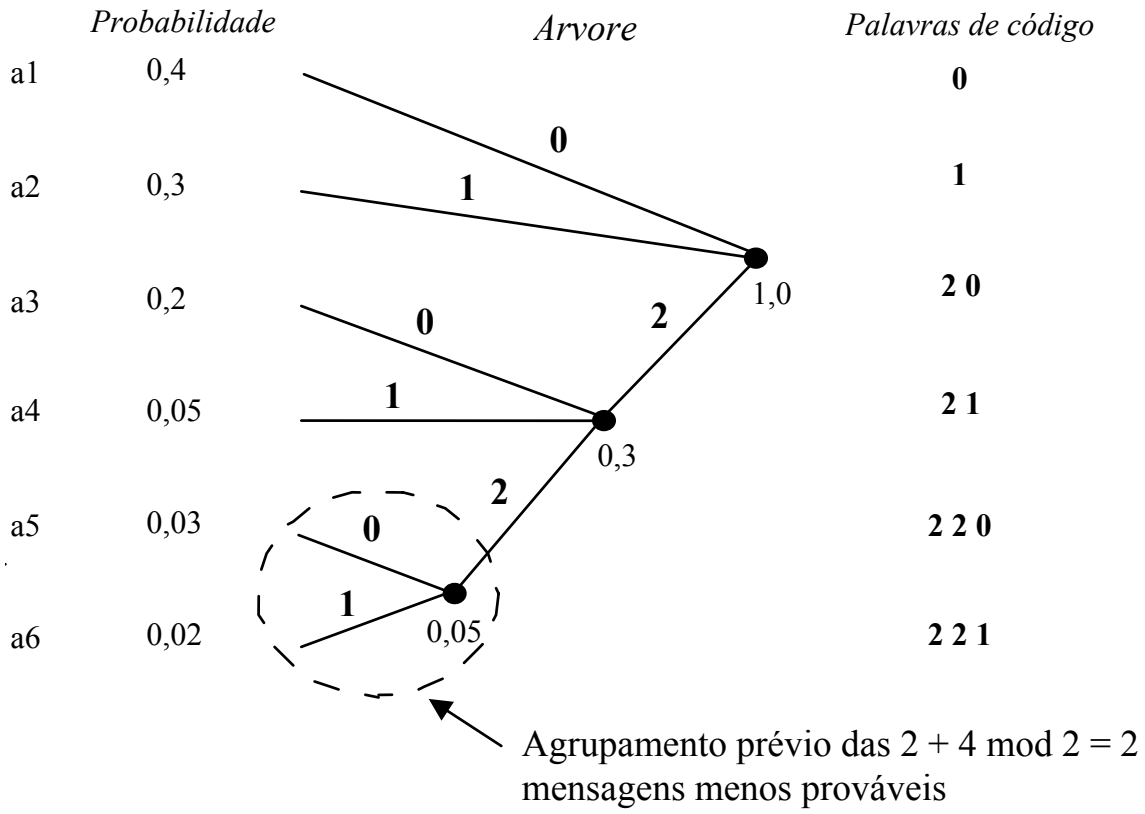
Repare-se que no código de Huffman a codificação se faz em direcção à raiz da árvore, ao contrário do código de Shannon-Fano – mas a formação das palavras de código faz-se no sentido contrário.

Código de Huffman não binário

Se o código não for binário ($D > 2$) é preciso proceder a um agrupamento inicial de mensagens de acordo com o seguinte procedimento[1]:

> Sendo M o tamanho do alfabeto da fonte e D o tamanho do alfabeto do código, agrupam-se primeiro as $2 + (M-2) \bmod (D-1)$ mensagens menos prováveis.
>
> ($(M-2) \bmod (D-1)$ é o resto da divisão de *M-2* por *D-1*)

EXEMPLO: *D* = 3, *M* = 6



O agrupamento prévio diminui o valor de $\overline{N}$ pois assim há menos palavras compridas (duas de 3 dígitos ternários em vez de três (D=3)).

[1] A justificação pode encontrar-se em R. G. Gallager, *Information Theory and Reliable Communication* (Wiley, 1968), Sec. 3.4.

Algo mais sobre o código de Huffman

Redução da variância

- Se ao formar um novo conjunto de probabilidades decrescentes houver probabilidades iguais as que resultam de agrupamentos devem ser colocadas o mais alto possível. Desse modo reduz-se a variância V dos comprimentos n_i das palavras de código. No entanto o comprimento médio $\overline{N}$ mantém-se.

$$\text{Variância} = V = E\left[n_i^2\right] - \overline{N}^2 \text{, com } E\left[n_i^2\right] = \sum_{i=1}^{M} P_i n_i^2$$

A variância mínima tem vantagens:

- o ritmo de produção de bits é mais uniforme (o que é conveniente para o preenchimento de "buffers");
- há uma maior imunidade (resistência) a erros do canal, na descodificação.

Aumento da eficiência da codificação não binária com agrupamento prévio

- O agrupamento prévio de símbolos na codificação de Huffman não-binária aumenta a sua eficiência pois baixa o comprimento médio $\overline{N}$, dado que assim as palavras mais compridas são em menor número.

Aplicações

- O código de Huffman é usado em equipamento comercial de compressão de dados para reduzir o número de bits transmitidos permitindo, portanto, reduzir o tempo de transmissão. Um exemplo concreto é o equipamento de *fax*.

Codificação aritmética

- Dizemos que um codificador é óptimo quando permite comprimir os dados gerados por uma determinada fonte com o número mínimo possível de bits.
- Apesar de os códigos de Huffman serem muito eficientes na maioria das aplicações, só são óptimos num reduzido número de casos (quando a probabilidade de ocorrência de cada símbolo for uma potência inteira de 1/2).
- A codificação aritmética surgiu em 1976 com Rissanen. Esta técnica de codificação permite gerar códigos óptimos se as diversas probabilidades de ocorrência forem conhecidas rigorosamente.

$$\Downarrow$$

É necessária uma correcta modelização da fonte
(como já acontecia na codificação de Huffman).

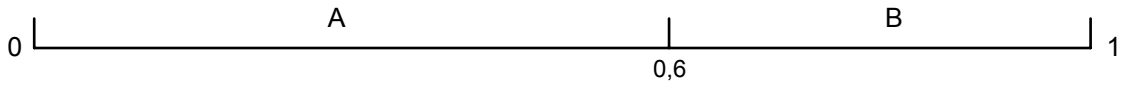
- O código gerado por um codificador aritmético representa um número real: o codificador faz corresponder a uma qualquer sequência de símbolos (cada um com a sua probabilidade) um número real superior ou igual a 0 e inferior a 1.
- O intervalo de 0 a 1 é inicialmente dividido em tantos intervalos quantos os símbolos diferentes produzidos pela fonte, de tamanhos proporcionais às probabilidades de ocorrência respectivas.
- Consoante o símbolo que for sendo gerado assim o respectivo intervalo vai sendo subdividido proporcionalmente.
- O resultado final será um número real incluído no intervalo final (fechado à esquerda e aberto à direita). O número escolhido deverá ser um de entre os que podem ser representados pelo menor número de bits (se a codificação for binária).
- É necessário acrescentar um símbolo especial ao conjunto de símbolos da fonte para assinalar o fim da sequência.

O melhor é vermos um exemplo…

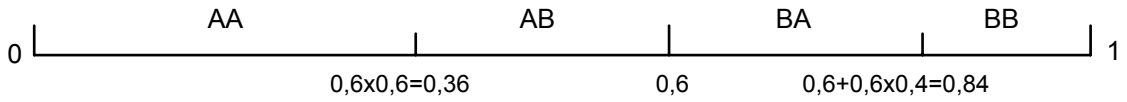
Codificação aritmética: introdução

Uma fonte produz os símbolos A e B com as probabilidades 0,6 e 0,4, respectivamente.

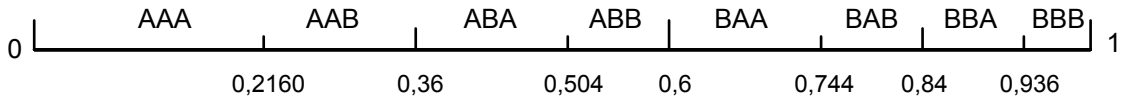
- Ao símbolo A vamos associar o intervalo semifechado de 0 a 0,6 (exclusivé) e a B vamos associar o intervalo semifechado de 0,6 a 1:



- Se cada um dos intervalos for dividido noutros dois com a mesma proporção (60% e 40%) obtemos quatro intervalos mais pequenos, $[0;0,36[$, $[0,36;0,6[$, $[0,6;0,84[$ e $[0,84;1[$:

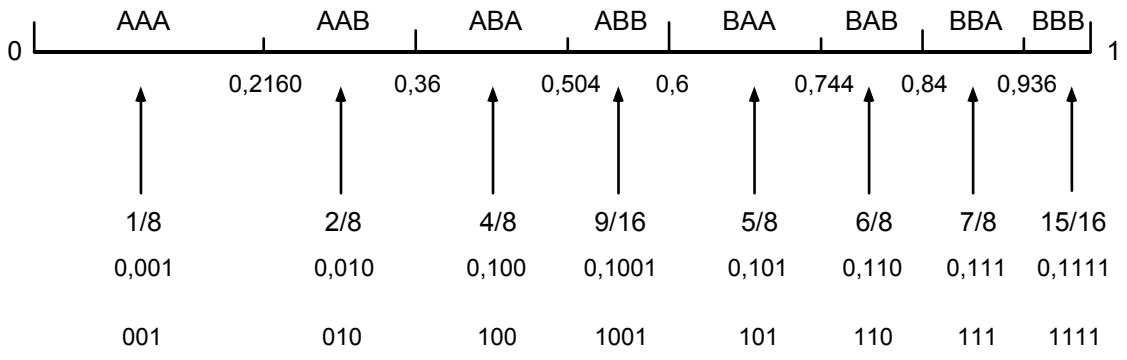


- Prosseguindo a subdivisão obteríamos oito intervalos ainda mais pequenos:



- À mensagem AAA corresponde o intervalo de 0 a 0,216 (com largura igual a $p(A)^3 = 0,216$), a BAB corresponde o intervalo de 0,744 até 0,84 (com largura $0,84 - 0,744 = p(B)p(A)p(B) = 0,096$), e por aí adiante.

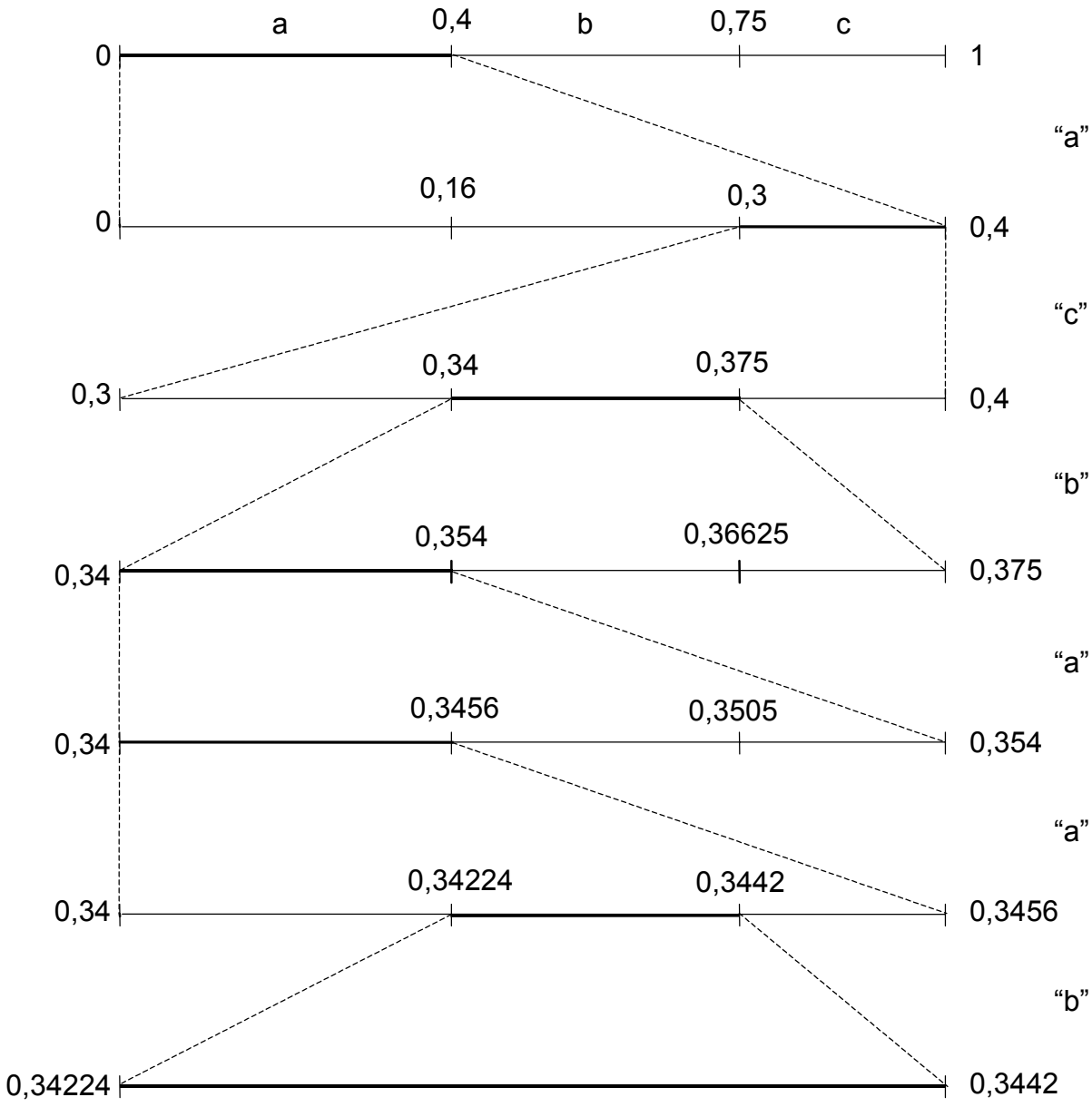
- Para codificar a mensagem AAA só temos que escolher um número dentro do intervalo correspondente, e de preferência um que seja representado pelo menor número de bits:



Codificação aritmética – exemplo 1

$P(a) = 0{,}4$ $P(b) = 0{,}35$ $P(c) = 0{,}25$

Sequência a codificar: a c b a a b



O código deve ser um número real incluído no intervalo [0,34224; 0,3442[.

Codificação aritmética – exemplo 2

Carácter	Probabilidade	Gama de valores
Espaço	0,1	[0,0 ; 0,1[
A	0,1	[0,1 ; 0,2[
B	0,1	[0,2 ; 0,3[
C	0,1	[0,3 ; 0,4[
E	0,1	[0,4 ; 0,5[
H	0,1	[0,5 ; 0,6[
L	0,2	[0,6 ; 0,8[
R	0,1	[0,8 ; 0,9[
V	0,1	[0,9 ; 1,0[

Codificação de “BARCA VELHA”:

Carácter	Valor inferior	Valor superior	Largura do intervalo
	0,0	1,0	1,0
B	0,2	0,3	10^{-1}
A	0,21	0,22	10^{-2}
R	0,218	219	10^{-3}
C	0,2183	0,2184	10^{-4}
A	0,21831	0,21832	10^{-5}
Espaço	0,21831	0,218311	10^{-6}
V	0,2183109	0,218311	10^{-7}
E	0,21831094	0,21831095	10^{-8}
L	0,218310946	0,218310948	$2x10^{-9}$
H	0,218310947	0,2183109472	$2x10^{-10}$
A	0,21831094702	0,21831094704	$2x10^{-11}$

“BARCA VELHA” = 0,21831094702 (por exemplo).

Procedimento de descodificação

1) Subtrair o valor inferior do intervalo da letra;
2) Dividir pela largura desse intervalo;
3) O resultado indica a nova letra;
4) Repetir

Exemplo abaixo:

$$0,21831094702 \Rightarrow B \rightarrow \frac{0,21831094702 - 0,2}{0,1} = 0,1831094702 \Rightarrow A$$

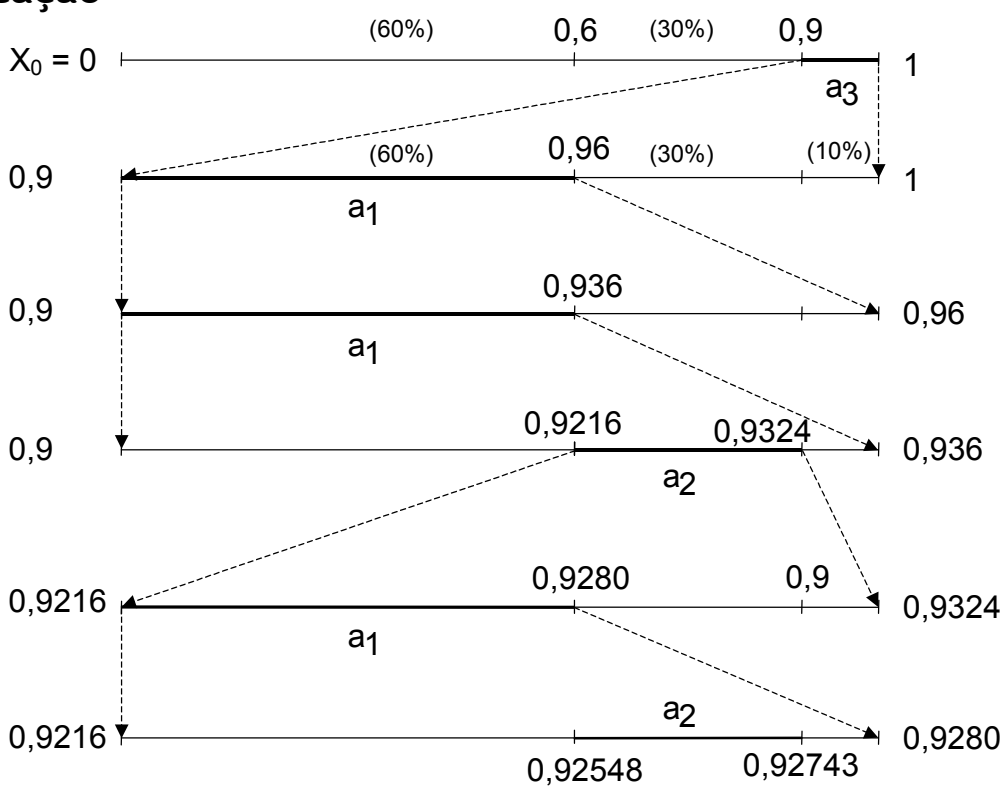
Descodificação de 0,21831094702:

Número codificado	Carácter	Valor inferior	Valor superior	Largura do intervalo
0,21831094702	B	0,2	0,3	0,1
0,1831094702	A	0,1	0,2	0,1
0,831094702	R	0,8	0,9	0,1
0,31094702	C	0,3	0,4	0,1
0,1094702	A	0,1	0,2	0,1
0,094702	Espaço	0,0	0,1	0,1
0,94702	V	0,9	1,0	0,1
0,4702	E	0,4	0,5	0,1
0,702	L	0,6	0,8	0,2
0,510	H	0,5	0,6	0,1
0,10	A	0,1	0,2	0,1
0,0				

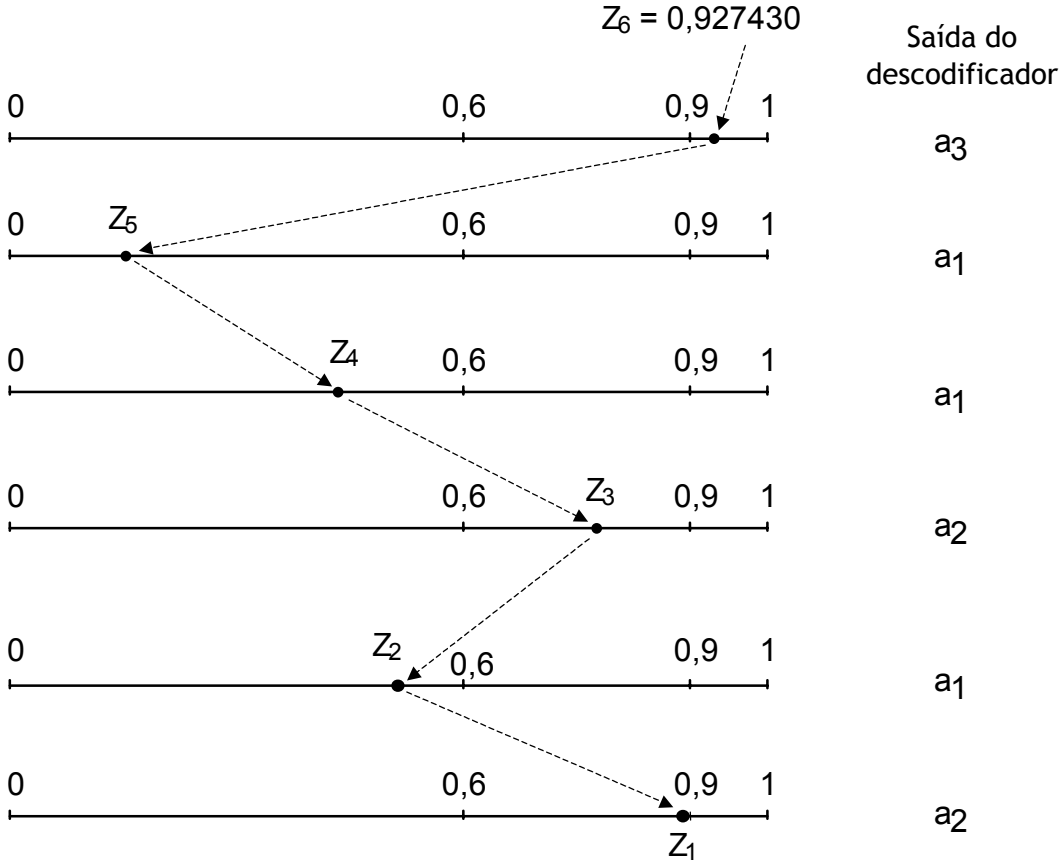
0,21831094702 = “BARCA VELHA”

Codificação aritmética – exemplo 3

Codificação



Descodificação



Inconvenientes dos códigos estatísticos

- É preciso conhecer a estatística da fonte. Para texto vulgar as frequências de ocorrência conhecidas bastariam; para texto mais especializado (reservas de avião, cotações da Bolsa, etc.) essas estatísticas não permitiriam uma codificação muito eficiente. Usando técnicas adaptativas consegue-se uma compressão de cerca de 43% em inglês vulgar* com código de Huffman.

- Em mensagens de texto a codificação não tem em conta a dependência entre letras (por exemplo, entre *q* e *u*) pois apenas são codificadas letras individuais.

- O facto das palavras de código terem comprimento variável põe problemas na transmissão pois a fonte está a gerar símbolos a uma taxa constante e o codificador produz símbolos a uma taxa variável. O emissor terá, pois, de possuir um *buffer* para obviar a este problema.

O código de Huffman é uma abordagem de codificação estatística. Uma alternativa que remedeia algumas das suas deficiências é a abordagem com *dicionários*, de que os algoritmos de *Lempel-Ziv* (L-Z) são exemplos. Nesses casos constrói-se gradualmente um dicionário, ou livro de código, isto é, usa-se um algoritmo adaptativo.

* R. Gitlin, J. Hayes e S. Weinstein, "Data Communications Principles", Plenum Press, 1992.

Codificação LZ77

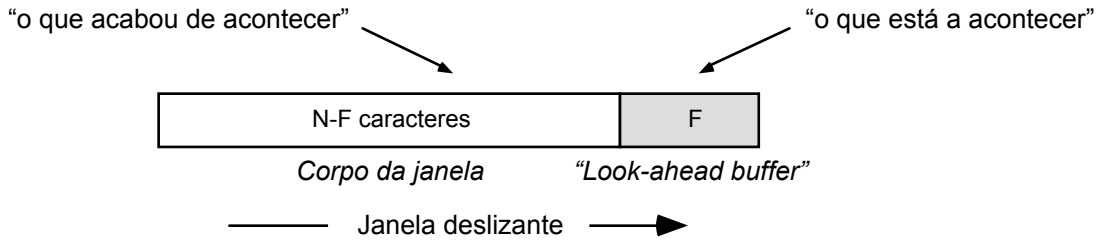
Usa-se uma janela de observação de comprimento fixo que desliza ao longo da mensagem a comprimir. É composta por duas partes:

- **corpo (da janela)**

  Contém vários milhares de caracteres (8192, por exemplo)

- **"look-ahead buffer"**

  Bastante mais pequeno (10 a 20 caracteres, ou até mais)



Passos a dar na codificação:

1. compara-se o conteúdo do corpo da janela com o conteúdo do *buffer* e verifica-se se existe algum padrão ou sequência de símbolos igual. Esse padrão é substituído na saída do codificador por um apontador constituído por um trio de elementos $< i, j, a >$ que na maior parte dos casos ocupa menos espaço que o padrão repetido:

i : distância (ou *offset*) do início do "buffer" ao início do padrão igual; especifica uma de $N - F$ posições na janela e pode ser 0.

j : comprimento do padrão; tem um valor entre 0 e $F - 1$.

a : primeiro carácter que já não faz parte do padrão repetido. É o próximo símbolo a codificar.

2. Desloca-se a janela para a frente de $j + 1$ caracteres.

A inclusão do próprio carácter *a* no apontador é para que a codificação prossiga mesmo que não haja padrão repetido. Nesse caso o apontador é $(0, 0, a)$.

LZ77: número de bits do apontador

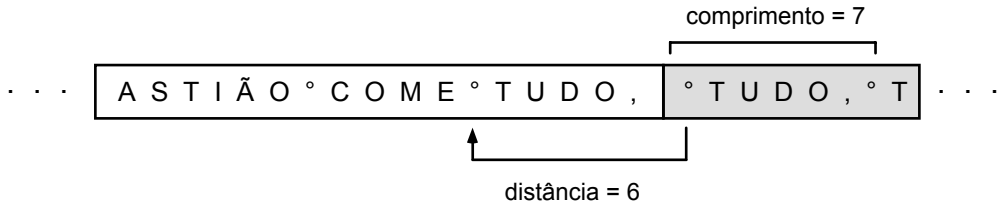
O número de bits necessários para representar cada apontador é sempre o mesmo:

- O primeiro elemento (*distância*) requer $\lceil \log_2 (N - F) \rceil$ bits;
- o segundo elemento (*comprimento*) requer $\lceil \log_2 F \rceil$ bits;
- o terceiro (*símbolo da fonte*) requer $\lceil \log_2 M \rceil$ bits, em que M é o tamanho do alfabeto da fonte.

Para representar cada apontador são precisos, no total,

$$\lceil \log_2 (N - F) \rceil + \lceil \log_2 F \rceil + \lceil \log_2 M \rceil \text{ bits}$$

- Exemplo com uma janela deslizante com $N = 24$ e $F = 8$



(o símbolo ◦ representa um espaço)

Ao olhar para o *look-ahead buffer* verifica-se que o padrão " ° TUDO,° ", de sete caracteres, já apareceu antes, concretamente seis caracteres atrás. Como a seguir vem o carácter *T*, " ° TUDO,° " será codificado como (6, 7, *T*).

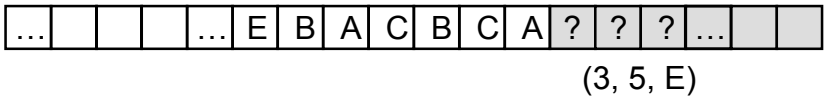
Número de bits de cada apontador (vamos supor que $M = 50$):

$$\lceil \log_2 (N - F) \rceil + \lceil \log_2 F \rceil + \lceil \log_2 M \rceil = \lceil \log_2 16 \rceil + \lceil \log_2 8 \rceil + \lceil \log_2 50 \rceil = 13$$

Descodificação LZ77

A janela deslizante tem o mesmo tamanho *N* que foi usado na codificação mas, em vez de se procurar uma correspondência entre corpo de janela e *look-ahead buffer* copia-se da própria janela para a saída do descodificador o padrão indicado pelo apontador.

Suponhamos que o apontador é (3, 5, *E)* quando, num determinado momento da codificação, a janela deslizante estiver preenchida como na figura:



Como descodificar o apontador (3, 5, *E)*?

A sequência …EBACBCA já está na saída do descodificador. Ora (3, 5, *E)* está a apontar para a terceira letra (a partir do fim) e corresponde a uma sequência de cinco letras seguida da letra E. Então na saída vão surgindo sucessivamente as seguintes letras:

1. …EBACBCA **B**
2. …EBACBCA B**C**
3. …EBACBCA B C **A**

Neste momento já temos três caracteres a partir do B inicial. Ainda faltam dois:

4. …EBACBCA B C A **B**
5. …EBACBCA B C A B **C**

Só falta o carácter seguinte, E:

6. …EBACBCA B C A B C **E**

Limitações de LZ77

1. Se no *look-ahead buffer* não houver nenhuma sequência que "case" com o conteúdo da janela o carácter a codificar é, mesmo assim, representado por um apontador constituído pelo trio de elementos habitual (distância, comprimento e carácter seguinte); nesse caso a codificação de um carácter necessitará de mais bits que o próprio carácter não codificado.

> Exemplo: se a janela tiver 4K bytes (de 8 bits) com um *buffer* de 16 bytes e se cada carácter não codificado for representado por 1 byte serão necessários 12 bits (*distância*) mais 4 bits (*comprimento*) mais 8 bits (*carácter*), isto é, 24 bits para representar um carácter que inicialmente só ocupava 8 bits: o codificador em vez de comprimir expandiu!!

2. A janela só observa o que é recente (porque o seu tamanho é finito): LZ77 é mais eficiente a comprimir padrões iguais próximos uns dos outros (isto é, que estejam na janela e no *buffer*) do que padrões iguais que estejam longe.

3. O tamanho limitado do *buffer* restringe o tamanho dos padrões que se comparam, que não podem ser maiores que o *buffer*. Mas às vezes as equivalências são mais compridas pelo que se perde eficiência.

Uma solução possível: aumentar os tamanhos da janela e do *buffer*, mas nesse caso serão precisos mais bits para representar a distância e o comprimento, além de que a codificação será mais lenta (mais comparações terão de ser feitas entre as sequências da janela e do *buffer*).

Há que estabelecer um compromisso entre, por um lado, os valores de N e F e, por outro, a *eficiência* e a *velocidade de compressão*.

Codificação LZ78

Existem imensas variantes do código LZ* e já conhecemos a LZ77. A variante que a seguir se apresenta, seguindo Gitlin *et al.*, é devida aos próprios "inventores", Jacob Ziv e Abraham Lempel (1978).

Na sequência de saída da fonte vamos procurar uma sequência de símbolos que ainda não tenha sido encontrada antes. Esta sequência é codificada usando sequências que já foram encontradas antes e que, por isso, já foram guardadas num *livro de código*, ou *dicionário*.

O algoritmo compreende-se melhor com um exemplo:

Suponhamos que a sequência original (na fonte) era

1 1 0 0 1 1 0 0 0 1 0 1 0 1 1 0 0 1 0 0 1 1 1 1 0 0 1

Percorrendo a sequência da esquerda para a direita e admitindo que no nosso livro de código já estão guardados os símbolos *0* e *1*, obteríamos a seguinte segmentação:

1 1 / 0 0 / 1 1 0 / 0 0 1 / 0 1 / 0 1 1 / 0 0 1 0 / 0 1 1 1 / 1 0 / 0 1

O livro de código, inicializado com os símbolos 0 e 1, vai conter os seguintes símbolos, com a numeração indicada:

0	1	11	00	110	001	01	011	0010	0111	10
1	2	3	4	5	6	7	8	9	10	11

A sequência a transmitir será a seguinte: **21 10 30 41 11 71 60 81 20 11.**

Cada inteiro é representado por um bloco de dígitos binários cujo tamanho depende do tamanho do livro de código. Na prática são normalmente usados blocos de 12 bits (isto é, 4096 entradas no livro de código).

Uma sequência nova termina sempre em 0 ou em 1. Consequentemente, apenas 1 bit é necessário para representar os fins de cada palavra de código, isto é, as palavras de código são representadas por 12+1=13 bits.

No algoritmo LZ78 palavras de código de comprimento fixo representam um número variável de símbolos de saída da fonte. Isto é o inverso do algoritmo de codificação de Huffman e é muito adequado para transmissão síncrona.

---

* Ver, por exemplo, T. Bell, J. Cleary e I. Witten, "Text Compression", Prentice-Hall, 1990.

Código LZ78

Exemplo

Consideremos a seguinte passagem literária:

"Foi ele que rapou avaramente a sopeira. E já espreitava a porta, esperando a portadora dos pitéus, a rija moça de peitos trementes, que enfim surgiu, mais esbraseada, abalando o sobrado—e pousou sobre a mesa uma travessa a transbordar de arroz com favas. Que desconsolo! Jacinto, em Paris, sempre abominara favas!…Tentou todavia uma garfada tímida—e de novo aqueles seus olhos, que o pessimismo enevoara, luziram, procurando os meus. Outra larga garfada, concentrada, com uma lentidão de frade que se regala. Depois um brado:

—Óptimo!…Ah, destas favas, sim! Oh que favas! Que delícia!"

Suponhamos que o livro de código já contém todas as letras, maiúsculas e minúsculas, espaço e todos os sinais de pontuação. A segmentação do texto seria assim:

"Fo/i /el/e /qu/e r/ap/ou/ a/va/ra/me/nt/e a/ s/op/ei/ra./ E/ j/á /es/pr/eit/av/a /a p/or/ta/, /esp/er/an/do/ a /po/rt/ad/ora/ d/os/ p/it/éu/s,/ a r/ij/a m/oç/a d/e p/eito/s /tr/em/en/te/s, /que/ e/nf/im/ su/rg/iu/, m/ai/s e/sb/ras/ea/da/, a/ba/la/nd/o /o s/ob/rad/o—/e po/us/ou /sob/re/ a m/esa/ u/ma/ t/rav/ess/a a/ tr/ans/bo/rd/ar/ de/ ar/ro/z /co/m /fa/vas/. /Qu/e d/esc/on/sol/o!/ J/ac/in/to/, e/m P/ari/s, s/emp/re /ab/om/ina/ra /fav/as/!…/Te/nto/u /tod/avi/a u/ma /ga/rf/ada/ tí/mi/da—/e d/e n/ov/o a/quel/es /se/us /ol/ho/s, q/ue/ o/ pe/ss/imi/sm/o e/ne/vo/ara/, l/uz/ir/am/, p/roc/ur/and/o o/s m/eu/s./ O/ut/ra l/arg/a g/arf/ada,/ c/onc/ent/rada/, c/om /um/a l/enti/dã/o d/e f/rade/ q/ue /se /reg/al/a./ D/ep/oi/s u/m b/rado/:

/—Ó/pt/imo/!…A/h,/ des/tas/ f/ava/s, si/m!/ Oh/ qu/e fa/vas!/ Q/ue d/elí/ci/a!/"

Cada nova sequência será codificada em duas palavras que já existem no livro de código, a última das quais faz parte do livro inicial. Repare-se, por curiosidade, como é codificada a palavra *favas*, que aparece quatro vezes.

Com o código LZ78 consegue-se uma compressão de aproximadamente 55% em inglês vulgar (em português não se conhecem dados). Em texto mais estruturado — linguagens de programação, por exemplo — pode-se atingir maiores compressões. Como já se disse antes, com o código de Huffman consegue-se chegar a 43%.

Codificação LZ78 – um exemplo

Mensagem a codificar: A A A A B A B C C A A B A C C A A A A B C

Entrada	Índice	Palavra de código
A	1	0A
B	2	0B
C	3	0C
AA	4	1A
AAB	5	4B
AB	6	1B
CC	7	3C
AABA	8	5A
CCA	9	7A
AAA	10	4A
BC	11	2C

Saída do codificador: 1A 4B 1B 3C 5A 7A 4A 2C

Compressão de dados

Programas ou formatos de compressão

Nome	Ambiente	Algoritmo	Comentário
COMPACT	Unix	Huffman adaptativo	Lento, patenteado
SQ	CP/M e MS-DOS	Huffman estático, ordem 0	Início anos 80; compressão comparável a COMPACT
ARC	PC	LZ78/compress	Comprime e arquiva
LHarc	PC/Mac	LZ77	“freeware”
ARJ	PC/Mac	LZ77	Melhor que LHarc
CPT	Mac	LZ	
DEFLATE		LZ77 + Huffman	Para substituir LZW. Muito popular. Autor: Phil Katz.
PKZIP, zip		DEFLATE	Autor: Phil Katz
gzip	Windows, Mac, Linux,…	LZ77	Para substituir *compress* Autores: Jean-Loup Gailly e Mark Adler
bzip		Codificação aritmética	Autor: Julian Seward
bzip2		Huffman + transformada de Burrows-Wheeler	Sucessor de bzip Autor: Julian Seward (1996)

(Continua)

Compressão de dados

Programas ou formatos de compressão
(Cont.)

Nome	Companhia	Algoritmo	Comentário
MNP-5	Microcom	variante dinâmica de Huffman	Modems
V42.bis	CCITT/ITU	LZW	Modems; melhor que MNP-5
G.721	CCITT/ITU	ADPCM	Compressão de voz
		LPC	Compressão de voz
StuffIt	Win, Mac, Linux	LZ	Utilitário de compressão
WinZip	Win, Mac, Linux	zip	Utilitário de compressão
GIF	Compuserve	LZW	Para imagens “bit-mapped”
JPEG	ISO/CCITT	DCT/Huffman	Para imagens fixas
MPEG	ISO/CCITT		Para imagens móveis
Adobe Acrobat	Adobe	LZW	Ficheiros .pdf

1.6. Fontes e canais contínuos

- entropia diferencial
- teorema da capacidade de canal (lei de Shannon-Hartley)
- limite de Shannon
- plano da probabilidade de erro
- plano da eficiência espectral

Fontes contínuas

- Uma fonte de informação *contínua* produz *sinais variáveis no tempo*, x(t).

- O conjunto de sinais possíveis é considerado como um *conjunto de formas de onda* geradas por um determinado *processo aleatório* (suposto ergódico).

Suponhamos que o processo aleatório tem uma largura de banda finita. Então x(t) pode ser completamente caracterizado em termos de valores amostra periódicos. Em qualquer instante a colecção de valores amostra possíveis constitui uma *variável aleatória X* descrita pela sua *função densidade de probabilidade*, p(x).

No caso das fontes contínuas a quantidade média de informação por amostra de x(t) mede-se através da função *entropia diferencial*:

$$h(X) = \int_{-\infty}^{\infty} p(x) \log_2 \frac{1}{p(x)}\, dx \qquad \text{(Entropia de fonte contínua)}$$

A entropia diferencial *h(X)* atinge o seu valor máximo se *p(x)* for uma função densidade de probabilidade gaussiana (ver Anexo 2)**:**

$$\max_{p_X(X)} h(X) = \frac{1}{2} \log_2 2\pi e S \qquad \text{em que } S = E(X^2) \text{ (variância)}$$

Fontes e canais contínuos

A transferência de informação num canal contínuo toma a forma de *transmissão de sinal*: a fonte emite um sinal $x(t)$ que, após corrupção por ruído de transmissão, aparece no destino como um outro sinal, $y(t)$.

A informação mútua média é definida como

$$I(X;Y) = \int_{-\infty}^{\infty} p_{XY}(x,y) \log_2 \frac{p_X(x|y)}{p_X(x)} dx\, dy$$

$I(X;Y)$ mede a transferência de informação por amostra de $y(t)$ no destino. Tal como no caso discreto também aqui se verifica que $I(X;Y) \geq 0$ (e $I(X;Y) = 0$ se o ruído for tão elevado que $y(t)$ não está relacionado com $x(t)$.

A informação mútua média está relacionada com a entropia diferencial através de

$$I(X;Y) = h(Y) - h(Y|X) = h(X) - h(X|Y)$$

em que *h(X)* representa a entropia diferencial da fonte, *h(Y)* representa a entropia diferencial do destino, $h(X|Y)$ é a *equivocação* e $h(Y|X)$ representa a entropia diferencial do "ruído", como se verá.

$$h(X|Y) = \int_{-\infty}^{\infty} p_{XY}(x,y) \log_2 \frac{1}{p_X(x|y)} dx\, dy = \int_{-\infty}^{\infty} p_Y(y) p_X(x|y) \log_2 \frac{1}{p_X(x|y)} dx\, dy$$

$$h(Y|X) = \int_{-\infty}^{\infty} p_{XY}(x,y) \log_2 \frac{1}{p_Y(y|x)} dx\, dy = \int_{-\infty}^{\infty} p_X(x) p_Y(y|x) \log_2 \frac{1}{p_Y(y|x)} dx\, dy$$

Capacidade de canais contínuos

Sabe-se, da teoria das probabilidades, que se $Y = aZ+b$, em que Z e Y são variáveis aleatórias, então $p_Y(y) = \frac{1}{a} p_Z\left(\frac{Y-b}{a}\right) = \frac{1}{a} p_Z(z)$. Assim, se o canal introduzir *ruído gaussiano aditivo independente*, tal que $y(t) = x(t) + n(t)$, então

$$p_Y(y \mid x) = p_Y(x+n) = p_N(y-x) = p_N(n)$$

e portanto

$$h(Y \mid X) = \int_{-\infty}^{\infty} \int p_X(x) p_Y(y \mid x) \log_2 \frac{1}{p_Y(y \mid x)} dx\, dy = \int_{-\infty}^{\infty} p_N(n) \log_2 \frac{1}{p_N(n)} dn = h(N)$$

Isto é, a entropia diferencial condicional de Y, dado X, não depende de $p_X(x)$ e é igual à entropia diferencial de N. Podemos, então escrever:

$$I(X;Y) = h(Y) - h(N)$$

A *transferência máxima de informação por amostra* de $y(t)$ obtém-se maximizando $I(X;Y)$. Esta maximização é equivalente à maximização de $h(Y)$, logo Y deve ser uma variável aleatória gaussiana. Como N também é gaussiana, X deve também ser gaussiana.

Se o canal tiver uma largura de banda fixa B então a sua saída, $y(t)$, é um sinal de banda limitada que pode ser completamente definido por amostras Y_k, $k = 1,2,\ldots$, obtidas à taxa de Nyquist, $f_s = 2B$ amostras/s. Definindo

$$C_s = \max_{p_X(x)} I(X;Y) = \left[\max I(X;Y) \ : \ \text{X gaussiana} \quad E(X^2) = S\right] \ \text{bits / amostra}$$

a *taxa máxima de transferência de informação*, em bits/s, é então

$C = 2BC_s$ bits/s (*capacidade de um canal contínuo de banda limitada*)

Capacidade do canal AWGN

Consideremos um canal contínuo de banda limitada afectado de ruído gaussiano branco aditivo (canal AWGN).

Propriedades:

— o sinal de entrada, $x(t)$, tem banda limitada e uma potência média fixa, $S = E(X^2)$.

— a transmissão não sofre distorção na largura de banda B do canal (a menos de atenuação…)

— o ruído aditivo que contamina a saída do canal tem função densidade de probabilidade gaussiana com média nula e potência média $N = N_0 B$

— o sinal e o ruído são independentes:

$$y(t) = x(t) + n(t) \;\Rightarrow\; E\left[Y^2\right] = \sigma_y^2 = S + N$$

- A entropia diferencial condicional $h(Y|X) = h(N)$ pode calcular-se a partir de $p_N(n)$, sendo $\sigma_n^2 = N$:

$$p_n(n) = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{n^2}{2\sigma_n^2}} \qquad \Rightarrow \qquad \log_2 \frac{1}{p_n(n)} = \log_2 \sqrt{2\pi N} + \frac{n^2}{2N} \log_2 e$$

$$\Rightarrow \quad h(Y|X) = h(N) = \int_{-\infty}^{\infty} p_n(n) \log_2 \frac{1}{p_n(n)}\, dn = \log_2 \sqrt{2\pi N} + \frac{1}{2}\log_2 e = \frac{1}{2}\log_2(2\pi e N)$$

- O valor máximo de $h(Y)$ é igual a $\frac{1}{2}\log_2 2\pi e\sigma_y^2 = \frac{1}{2}\log_2 2\pi e(S+N)$

(Continua)

Capacidade do canal AWGN

(Contin.)

Podemos então calcular a *capacidade de um canal AWGN*:

- $h(Y|X) = \frac{1}{2}\log_2(2\pi eN)$

- $\max\limits_{p_X(x)} h(Y) = \frac{1}{2}\log_2 2\pi e(S+N)$

- $C_s = \max\limits_{p_X(x)} I(X;Y) = \max\limits_{p_X(x)}\left[h(Y)\right] - h(Y|X) =$

  $= \frac{1}{2}\log_2 2\pi e(S+N) - \frac{1}{2}\log_2 2\pi eN = \frac{1}{2}\log_2\left(1+\frac{S}{N}\right)$ bits / amostra

- *Capacidade do canal* em bits/s:

$$C = B\log_2\left(1+\frac{S}{N}\right)$$

Esta expressão traduz o *Teorema da capacidade de canal* ou *lei de Shannon-Hartley*, que é um dos mais famosos e importantes resultados da Teoria da Informação:

Não é possível transmitir sem erros a um ritmo superior a $C = B\log_2\left(1+\frac{S}{N}\right)$ *bits/s através de um canal com largura de banda B e relação sinal-ruído S/N.*

Implicações do teorema da capacidade de canal

- Um sistema *ideal* com largura de banda *infinita* tem uma capacidade de canal *finita*:

$$C = B\log_2\left(1+\frac{S}{N}\right) = B\log_2\left(1+\frac{S}{N_0 B}\right) =$$

$$= \frac{B}{\ln 2}\ln\left(1+\frac{S}{N_0 B}\right)$$

Se $B \to \infty,\ S/N_0 B \to 0$. Mas [ ]. Portanto,

$$C_\infty = \lim_{B\to\infty} C \approx \frac{B}{\ln 2}\frac{S}{N_0 B} = \frac{1}{\ln 2}\frac{S}{N_0} = 1{,}44\frac{S}{N_0}$$

- Existe um valor limite de E_b/N_0 *(limite de Shannon)* abaixo do qual não pode haver comunicação sem erros, qualquer que seja o ritmo de transmissão.

Sendo T a duração de cada bit, E_b a sua energia e $R = 1/T$ bits/s, então

$$\frac{S}{N} = \frac{E_b/T}{N_0 B} = \frac{E_b}{N_0}\frac{R}{B} \qquad \Rightarrow \qquad C = B\log_2\left(1+\frac{S}{N}\right) = B\log_2\left(1+\frac{E_b}{N_0}\frac{R}{B}\right)$$

Mas $R \le C \quad \Rightarrow \quad \frac{R}{B} \le \frac{C}{B} = \log_2\left(1+\frac{E_b}{N_0}\frac{R}{B}\right)$

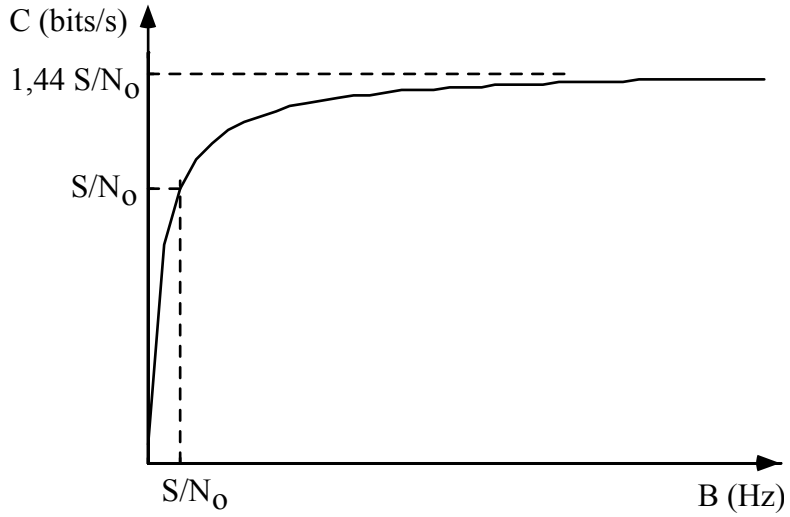
Se $B \to \infty \quad \Rightarrow \qquad \frac{R}{B} \le \log_2\left(1+\frac{E_b}{N_0}\frac{R}{B}\right) = \frac{\ln\left(1+\frac{E_b}{N_0}\frac{R}{B}\right)}{\ln 2} \approx \frac{\frac{E_b}{N_0}\frac{R}{B}}{\ln 2}$

Portanto,

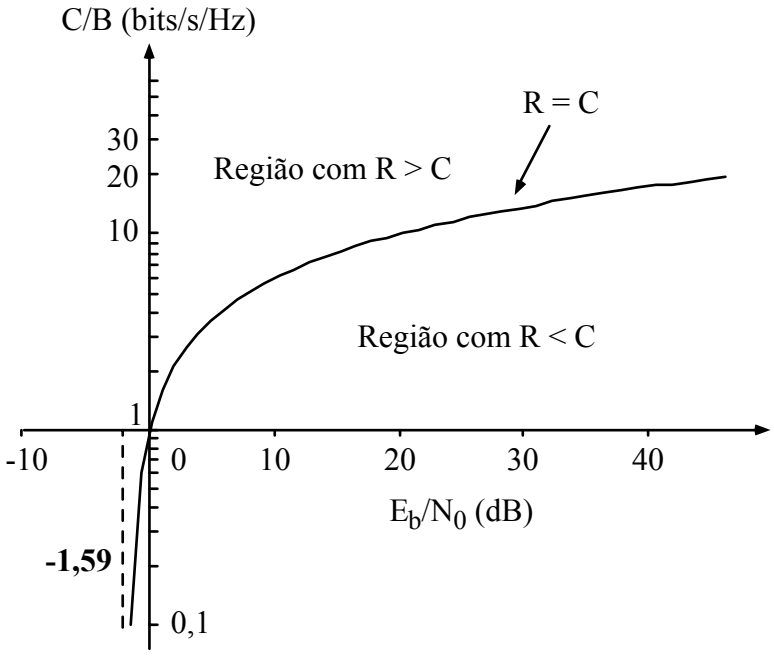
$$\frac{E_b}{N_0} \ge \ln 2 = 0{,}693 \to -1{,}59\,dB \qquad \textit{(limite de Shannon)}$$

Capacidade de canais contínuos

Capacidade do canal em função da largura de banda



O limite de Shannon



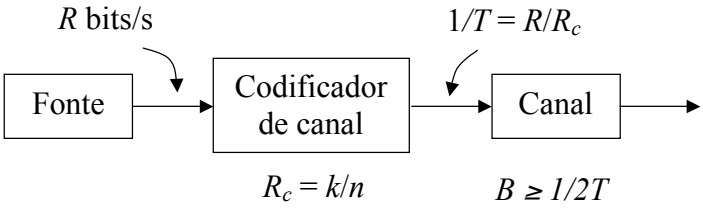
Capacidade de canal AWGN com largura de banda finita

Temos $\frac{R}{B} \le \frac{C}{B} = \log_2\left(1+\frac{S}{N}\right) = \log_2\left(1+\frac{E_b}{N_0}\frac{R}{B}\right)$, donde se tira

$$\frac{E_b}{N_0} \ge \frac{2^{R/B}-1}{R/B}$$

(R/B é a eficiência espectral)

Consideremos agora a seguinte situação com *codificação de canal*:



Para que não haja interferência intersimbólica é preciso que $B \ge 1/2T$, ou seja,

$$B \ge \frac{1}{2T} = \frac{R}{2R_c} \qquad \text{ou} \qquad \frac{R}{B} \le 2R_c$$

Se queremos aumentar a eficiência espectral teremos de aumentar a taxa do código.

Assim: $$\frac{E_b}{N_0} \ge \frac{2^{2R_c}-1}{2R_c}$$

Exemplos:

$R_c = 1/2 \quad \Rightarrow \quad \frac{E_b}{N_0} \ge 0\ dB$

$R_c = 1/3 \quad \Rightarrow \quad \frac{E_b}{N_0} \ge -0{,}55\ dB$

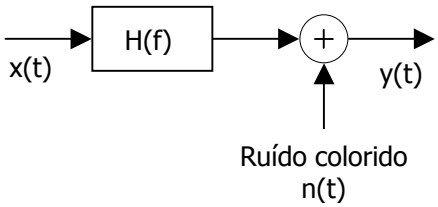
$R_c = 1/4 \quad \Rightarrow \quad \frac{E_b}{N_0} \ge -0{,}82\ dB$

À medida que a taxa de código diminui mais nos podemos aproximar do limite de Shannon (-1,59 dB).

Capacidade de um canal afectado de ruído gaussiano colorido

O "enchimento de água"

Vamos tratar do caso mais geral de ruído colorido.



- Admitamos que o sinal de entrada $x(t)$ tem uma potência média P fixa e que $S_X(f)$ é a sua densidade espectral de potência:

$$P = \int_{-\infty}^{\infty} S_X(f)df$$

- O ruído gaussiano estacionário colorido tem média nula e $S_N(f)$ é a sua densidade espectral de potência.
- $H(f)$ é a função de transferência do canal.

Qual é a capacidade deste canal? E qual é a densidade espectral de potência do sinal de entrada, $S_X(f)$, que maximiza a informação mútua média?

Os resultados, obtidos com o *método dos multiplicadores de Lagrange*, são os seguintes, sendo K uma constante:

$$S_X(f) = \begin{cases} K - \dfrac{S_N(f)}{|H(f)|^2} & f \in F \\ 0 & outros \end{cases}$$

$$C = \frac{1}{2}\int_{-\infty}^{\infty} \log_2\left(K\frac{|H(f)|^2}{S_N(f)}\right)df$$

Capacidade de um canal afectado de ruído gaussiano colorido

O "enchimento de água"

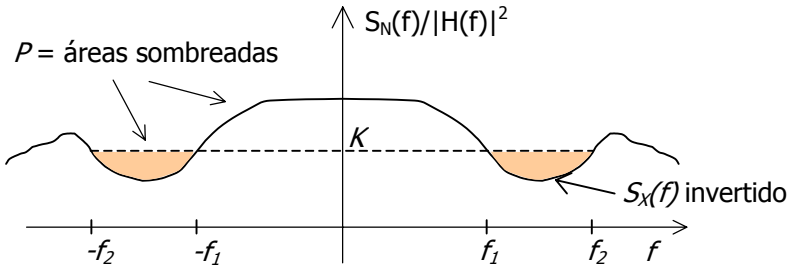
(cont.)

Para valores especificados de P, $S_N(f)$ e $H(f)$ a constante K é a solução da equação seguinte:

$$P = \int_{f \in F} S_X(f)df = \int_{f \in F} \left(K - \frac{S_N(f)}{|H(f)|^2} \right) df$$

O conjunto $F = \{f : f_1 \le |f| \le f_2\}$ representa a gama de valores de frequência para os quais a constante K é superior a $\frac{S_N(f)}{|H(f)|^2}$.

Estas considerações conduzem à chamada *"interpretação do enchimento de água"*:



É como se estivéssemos a "encher" de água (ou melhor, de potência de $x(t)$) o gráfico de $S_N(f)/|H(f)|^2$: onde houver "menos" ruído (nos vales) coloca-se mais potência de sinal de entrada, onde houver mais ruído coloca-se menos potência.

Capacidade de um canal afectado de ruído gaussiano colorido

Exemplo 1: Canal com ruído AWGN (o ruído é branco)

$$S_N(f) = \frac{N_0}{2}$$

$$H(f) = \begin{cases} 1 & 0 \le f_c - B/2 \le |f| \le f_c + B/2 \\ 0 & outros \end{cases} \qquad \text{(canal ideal)}$$

Da expressão da potência, $P = 2\int_{f_c - B/2}^{f_c + B/2} (K - N_0/2)df$, tiramos o valor de K:

$$K = \frac{P + N_0 B}{2B}$$

A capacidade deste canal obtém-se imediatamente:

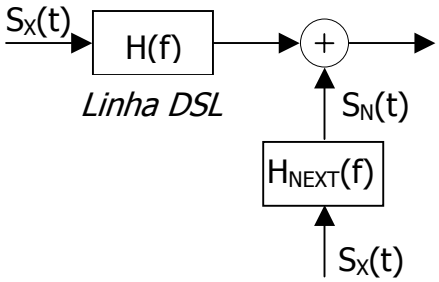
$$C = 2 \times \frac{1}{2} \int_{f_c - B/2}^{f_c + B/2} \log_2 K \frac{|H(f)|^2}{S_N(f)} df = B \log_2 \frac{2K}{N_0}$$

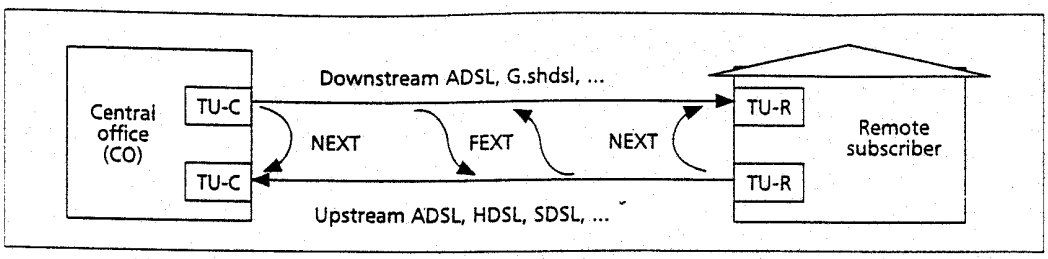
Substituindo o valor de K obtido atrás chegamos à expressão já conhecida do teorema de Shannon-Hartley:

$$C = B \log_2 \frac{2\dfrac{P + N_0 B}{2B}}{N_0} = B \log_2 \left(1 + \frac{P}{N_0 B} \right)$$

Capacidade de um canal afectado de ruído gaussiano colorido

Exemplo 2: DSL com interferência NEXT ("near-end crosstalk")





NEXT and FEXT.

- Normalmente $H_{NEXT}(f) = \beta f^{3/2}$, em que β é uma característica dos cabos telefónicos.
- $S_N(f) = |H_{NEXT}(f)|^2 S_X(f)$

A única restrição a impor é que a densidade espectral de potência de $x(t)$ seja não-negativa. Substituindo valores chega-se à constante *K*,

$$K = \left(1 + \frac{|H_{NEXT}(f)|^2}{|H(f)|^2}\right) S_X(f),$$

e daí à capacidade do lacete de assinante (DSL) afectado de "crosstalk" do tipo NEXT:

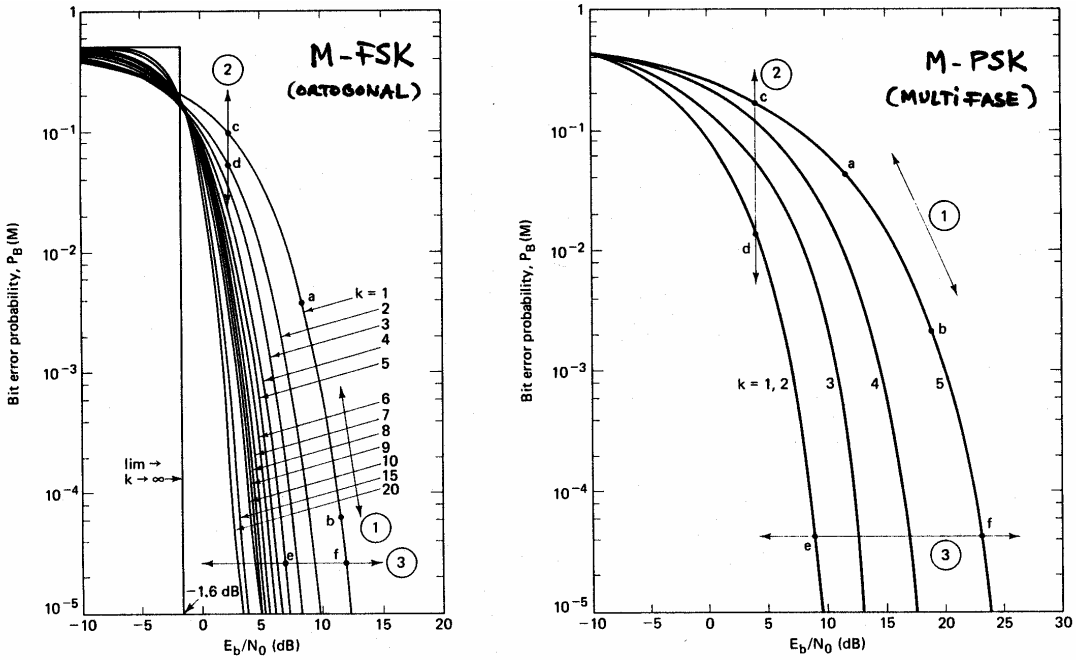
$$C = \frac{1}{2} \int_{f \in F} \log_2 \left(1 + \frac{|H(f)|^2}{|H_{NEXT}(f)|^2}\right) df$$

(*F* é o conjunto das frequências positivas e negativas para as quais $S_X(f) > 0$)

Plano da probabilidade de erro

- É o *lugar geométrico* dos pontos de funcionamento disponíveis para um determinado tipo de modulação ou codificação.

- Cada curva está associada à largura de banda mínima exigida.



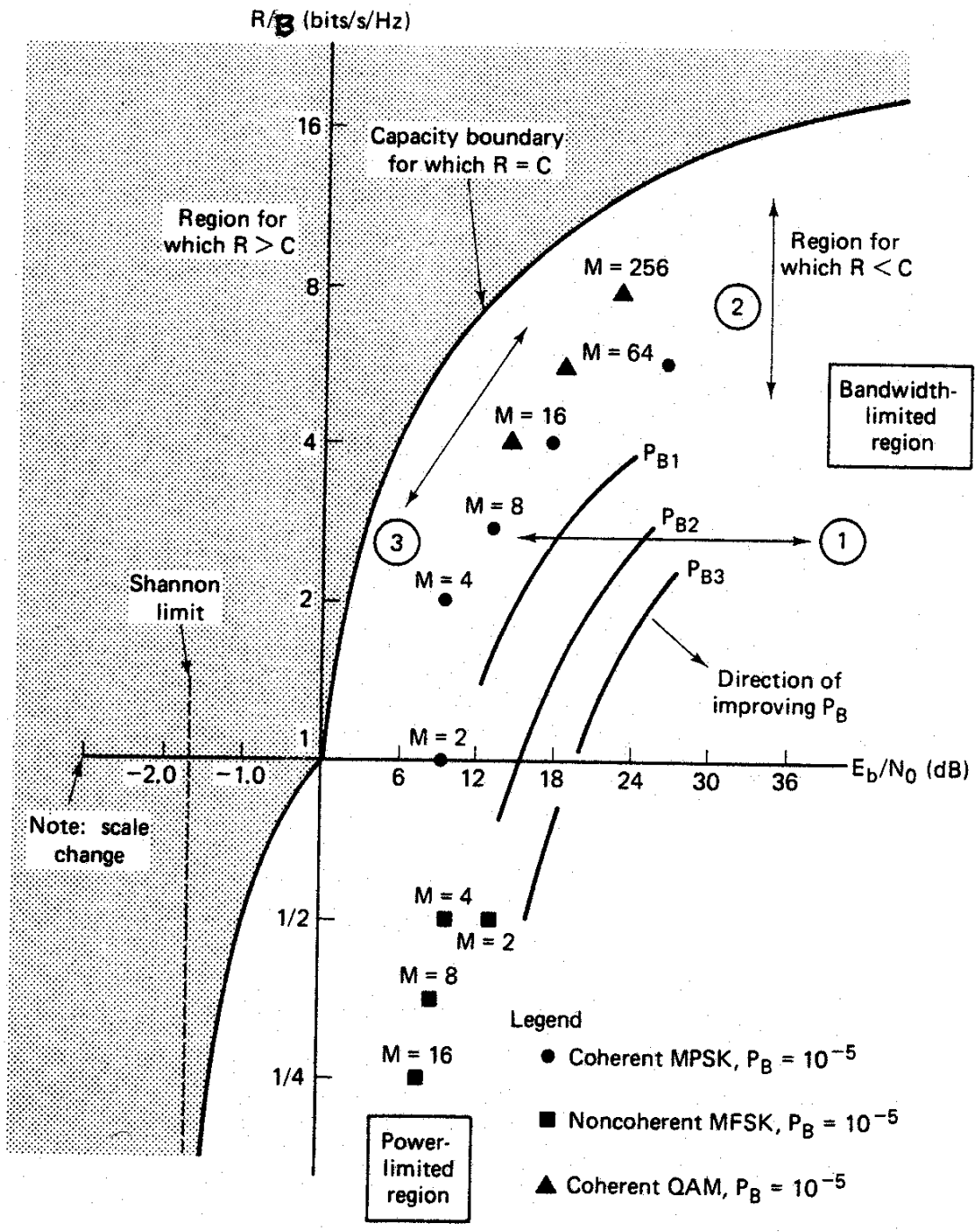
O funcionamento do sistema é caracterizado por um ponto no plano da probabilidade de erro, para um determinado tipo de modulação escolhido e para a razão E_b/N_O disponível.

<u>Compromissos:</u>

- Em funcionamento: 1 – Entre P_B e E_b/N_0 (com B fixo)
- Só na fase de projecto 2 – Entre P_B e B (com E_b/N_0 fixo)
  3 – Entre B e E_b/N_0 (com P_B fixo)

O compromisso 1 só envolve variação de potência. Os restantes envolvem mudanças nos tipos de modulação e codificação.

Capacidade de canal gaussiano e eficiência espectral de modulações



Anexo 1

Desigualdade de Kraft (ou de Kraft-McMillan)

Teorema: Se os inteiros $n_1, n_2, ..., n_M$ satisfazem a desigualdade

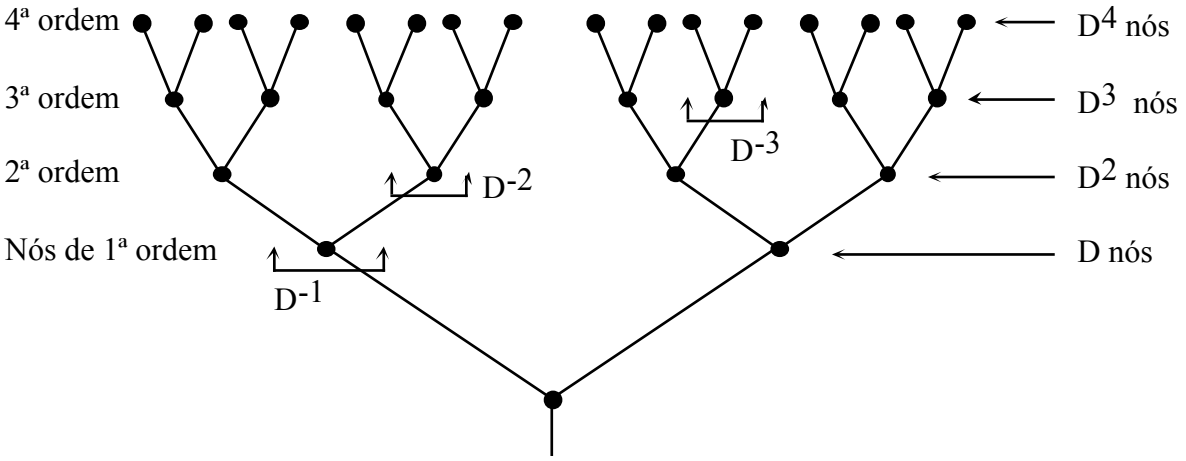
$$\sum_{i=1}^{M} D^{-n_i} \leq 1$$

então existe um código sem prefixos com um tamanho de alfabeto D cujos comprimentos de palavras são esses inteiros.

Inversamente: os comprimentos de qualquer código sem prefixos satisfazem a desigualdade.

O teorema *não diz* que qualquer código cujos comprimentos satisfazem a desigualdade é um código sem prefixos. O que o teorema diz é que existe um certo código sem prefixos e que tem esses comprimentos.

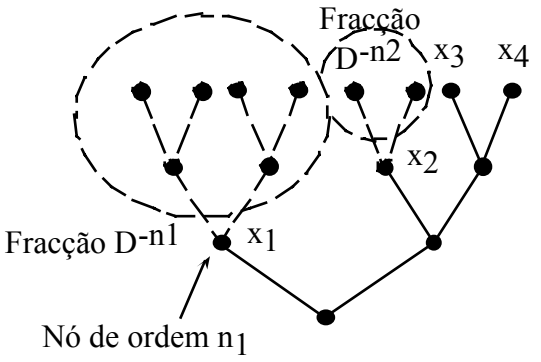
Arvore cheia (com D=2)



- Existem D^n nós de ordem n
- De cada nó de ordem i saem D ramos ligados a D nós de ordem i+1.
- De cada um dos D nós de ordem 1 sai uma fracção D^{-1} dos nós de ordem $i \geq 1$.
- De cada um dos D^2 nós de ordem 2 sai uma fracção D^{-2} dos nós de ordem $i \geq 2$.

Genericamente: De cada um dos $\boldsymbol{D^i}$ nós de ordem $\boldsymbol{i}$ sai uma fracção $\boldsymbol{D^{-i}}$ dos nós de ordem maior ou igual a $\boldsymbol{i}$.

Desigualdade de Kraft



Sejam $n_1 \leq n_2 \leq ... \leq n_M$ um conjunto de M inteiros que satisfazem a desigualdade de Kraft.

Suponhamos que a árvore ao lado tem ordem n igual ao maior inteiro n_M.

Demonstração do teorema:

1 — Tomemos um nó qualquer de ordem n_1 (x_1, por exemplo) como o primeiro nó terminal da árvore de código. Todos os nós de cada ordem $\geq n_1$ estão ainda disponíveis para nós terminais,

excepto uma fracção D^{-n_1} (nós que saem de x_1)

2 — A seguir tomemos outro qualquer nó de ordem n_2 (x_2, por exemplo). Todos os nós de cada ordem $\geq n_2$ estão ainda disponíveis para nós terminais,

excepto a fracção $D^{-n_1} + D^{-n_2}$ (nós que saem de x_1 e x_2)

3 — Continuando dessa maneira, depois da atribuição do nó terminal de ordem k na árvore de código, todos os nós da árvore cheia, de cada ordem $\geq n_k$, estão ainda disponíveis

excepto a fracção $D^{-n_1} + D^{-n_2} + \ldots + D^{-n_k} = \sum_{i=1}^{k} D^{-n_i}$

4 — Por hipótese $\sum_{i=1}^{k} D^{-n_i} < 1$, para $i < M \Rightarrow$ há sempre um nó disponível para servir como nó terminal $\Rightarrow$ existe um código sem prefixos.

Inversamente: A árvore de código de um código sem prefixos está contida numa *árvore cheia* $\Rightarrow$ um nó terminal de ordem n_i dá origem a uma fracção D^{-n_i} dos nós terminais da árvore cheia. Como os conjuntos dos nós terminais da árvore cheia que derivam dos nós terminais da árvore de código são disjuntos, conclui-se que essas fracções somadas são ≤ 1, *c.q.d.*

Anexo 2

Valor máximo da entropia diferencial

Teorema

O valor máximo da entropia diferencial $h(Y) = -\int_{-\infty}^{\infty} p(y)\log_2 p(y)\,dy$ para todas as escolhas de funções densidade de probabilidade satisfazendo

$$\int_{-\infty}^{\infty} y^2 p(y)\,dy = \sigma^2 \tag{a}$$

é **unicamente** atingido pela função densidade de probabilidade gaussiana

$$\varphi(y) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{y^2}{2\sigma^2}} \tag{b}$$

e tem o valor $h(Y) = \frac{1}{2}\log_2(2\pi e\sigma^2)$.

Demonstração (de acordo com R. G. Gallager, *Information Theory and Reliable Communication* (Wiley, 1968))

Seja $p(y)$ uma fdp arbitrária satisfazendo (a) e seja $\varphi(y)$ a fdp gaussiana de (b). Então

$$\int_{-\infty}^{\infty} p(y)\log_2 \frac{1}{\varphi(y)}\,dy = \int_{-\infty}^{\infty} p(y)\left[\log_2\sqrt{2\pi\sigma^2} + \frac{y^2}{2\sigma^2}\log_2 e\right]dy =$$

$$= \log_2\sqrt{2\pi\sigma^2} + \frac{\sigma^2}{2\sigma^2}\log_2 e = \frac{1}{2}\log_2(2\pi e\sigma^2)$$

Portanto,

$$h(Y) - \frac{1}{2}\log_2(2\pi e\sigma^2) = \int_{-\infty}^{\infty} p(y)\log_2\frac{\varphi(y)}{p(y)}\,dy$$

Como $\log_2 z \le (z-1)\log_2 e$, então

$$h(Y) - \frac{1}{2}\log_2(2\pi e\sigma^2) \le \log_2 e \int_{-\infty}^{\infty} p(y)\left[\frac{\varphi(y)}{p(y)} - 1\right]dy = 0$$

A igualdade obtém-se apenas **sse** $\frac{\varphi(y)}{p(y)} = 1$ para todos os y, *c.q.d.*

Resumo do Capítulo 1

Neste capítulo foram estabelecidos três limites fundamentais em diferentes aspectos de um sistema de comunicações digitais através do *teorema da codificação da fonte*, do *teorema da codificação de canal* e do *teorema da capacidade do canal*, todos devidos a Claude Shannon.

O *teorema da codificação da fonte* fornece-nos um padrão pelo qual podemos medir a informação proveniente de uma *fonte discreta sem memória*. Diz-nos o teorema que podemos tornar o número médio de elementos binários (binits) por símbolo da fonte tão pequeno como, mas não menor que, a *entropia* da fonte medida em bits. A entropia de uma fonte é uma função das probabilidades dos símbolos que a fonte produz. Visto a entropia ser uma medida de incerteza, é máxima quando a distribuição de probabilidades associadas gera a máxima incerteza, ou seja, quando os símbolos são equiprováveis.

Os objectivos do teorema da codificação da fonte são perseguidos e aproximados se, antes da codificação, os símbolos produzidos pela fonte forem agrupados em blocos de N símbolos – teremos então uma *extensão de ordem N* – e se os códigos forem de *comprimento variável*. Destes só nos interessam *códigos unicamente descodificáveis*, em especial um seu subconjunto, o dos *códigos sem prefixos*, definidos através de *árvores*. Ambos obedecem à *desigualdade de Kraft-McMillan*, que relaciona entre si os comprimentos de cada palavra de código. Os códigos de *Shannon-Fano* e *Huffman* são dois exemplos de códigos sem prefixos cujo comprimento médio das palavras de código se aproxima da entropia da fonte.

Uma outra abordagem foi feita através da *codificação aritmética*, embora também baseada em modelos probabilísticos da fonte, tal como os dois códigos precedentes; completamente diferente são as codificações de *Lempel-Ziv* (LZ77 e LZ78). Nesta última, por exemplo, vai-se construindo um dicionário à medida que a codificação se realiza.

O *teorema da codificação de canal* é o resultado mais surpreendente e talvez o mais importante da Teoria da Informação. Por exemplo, para um *canal binário simétrico* (BSC), o teorema da codificação de canal diz-nos que para qualquer *taxa de informação* menor ou igual à *capacidade do canal* existem *códigos* tais que a probabilidade de erro média é tão pequena quanto quisermos. Um canal binário simétrico é a forma mais simples de um *canal discreto sem memória*; é simétrico se a probabilidade de se receber um 1 se se tiver enviado um 0 for igual à probabilidade de se receber um 0 se se tiver enviado um 1. Esta probabilidade, chamada *probabilidade de transição*, define univocamente a capacidade do canal.

O terceiro teorema de Shannon, provavelmente o mais famoso, é o *teorema da capacidade do canal*, ou *lei de Shannon-Hartley*, e aplica-se a canais contínuos. Diz-nos o teorema que em qualquer canal de comunicação limitado em potência existe uma velocidade máxima de transmissão de informação sem erros, dependente da largura de banda do canal e da relação sinal-ruído. Esta velocidade, ou taxa, máxima chama-se *capacidade do canal* e mede-se em bits/s. Se o sistema

funcionar a uma taxa superior à capacidade do canal está condenado a uma probabilidade de erro elevada, independentemente do receptor ou da técnica de transmissão usada.

A codificação de canal (ou codificação com controlo de erros), referida no teorema da codificação de canal, vai ser objecto do nosso estudo em seguida.