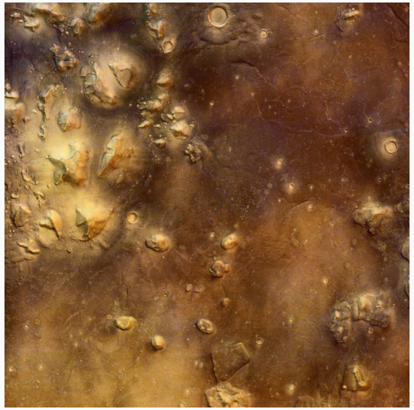
Sistemas de Comunicação II

Teoria da informação



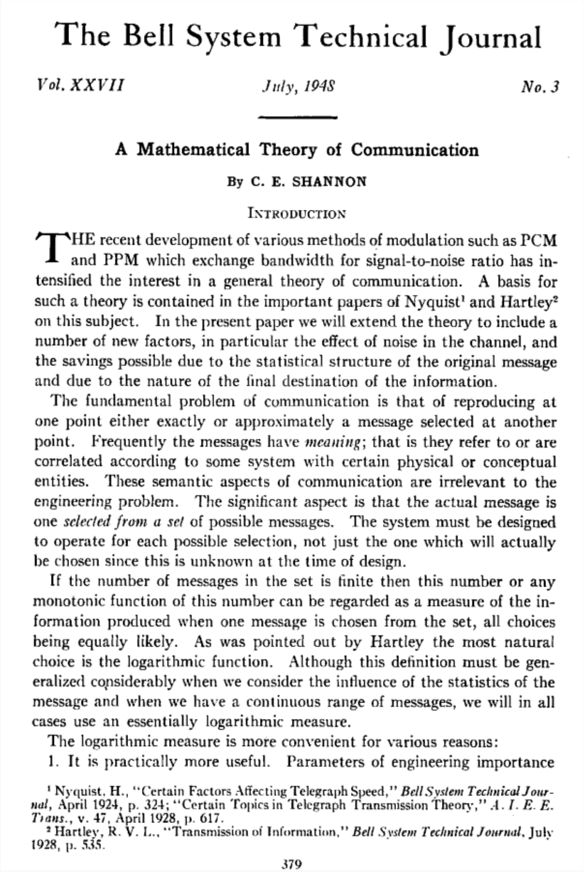
Prof. Roberto Wanderley da Nóbrega

Instituto Federal de Santa Catarina

Introdução

Uma teoria matemática da comunicação

Shannon: *A Mathematical Theory of Communication*. Bell Labs, 1948.



The Bell System Technical Journal

Vol. XXVII *July, 1948* *No. 3*

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[1] and Hartley[2] 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.

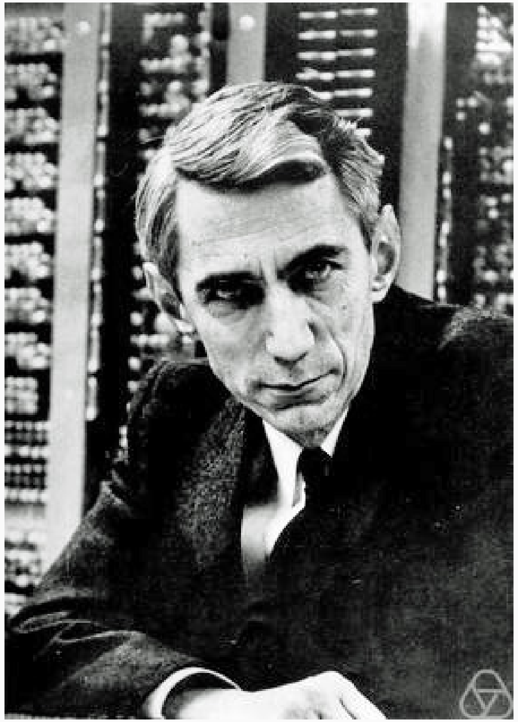
If the number of messages in the set is finite then this number or any monotonic function of this number can be regarded as a measure of the information produced when one message is chosen from the set, all choices being equally likely. As was pointed out by Hartley the most natural choice is the logarithmic 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

[1] Nyquist, H., "Certain Factors Affecting Telegraph Speed," *Bell System Technical Journal*, April 1924, p. 324; "Certain Topics in Telegraph Transmission Theory," *A. I. E. E. Trans.*, v. 47, April 1928, p. 617.
[2] Hartley, R. V. L., "Transmission of Information," *Bell System Technical Journal*, July 1928, p. 535.

379



Codificação de fonte

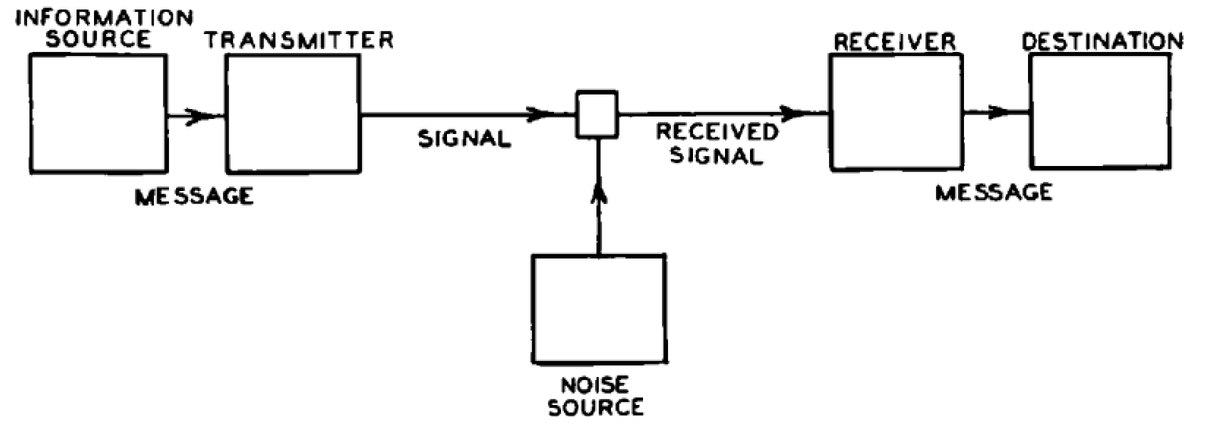
Representar com **fidelidade** uma *fonte de informação* através de uma *sequência de bits* com a **menor taxa possível**.

Conceito chave: **Entropia da fonte**.

Codificação de canal

Transmitir com **confiabilidade** uma *sequência de bits* por um *canal de comunicação* com a **maior taxa possível**.

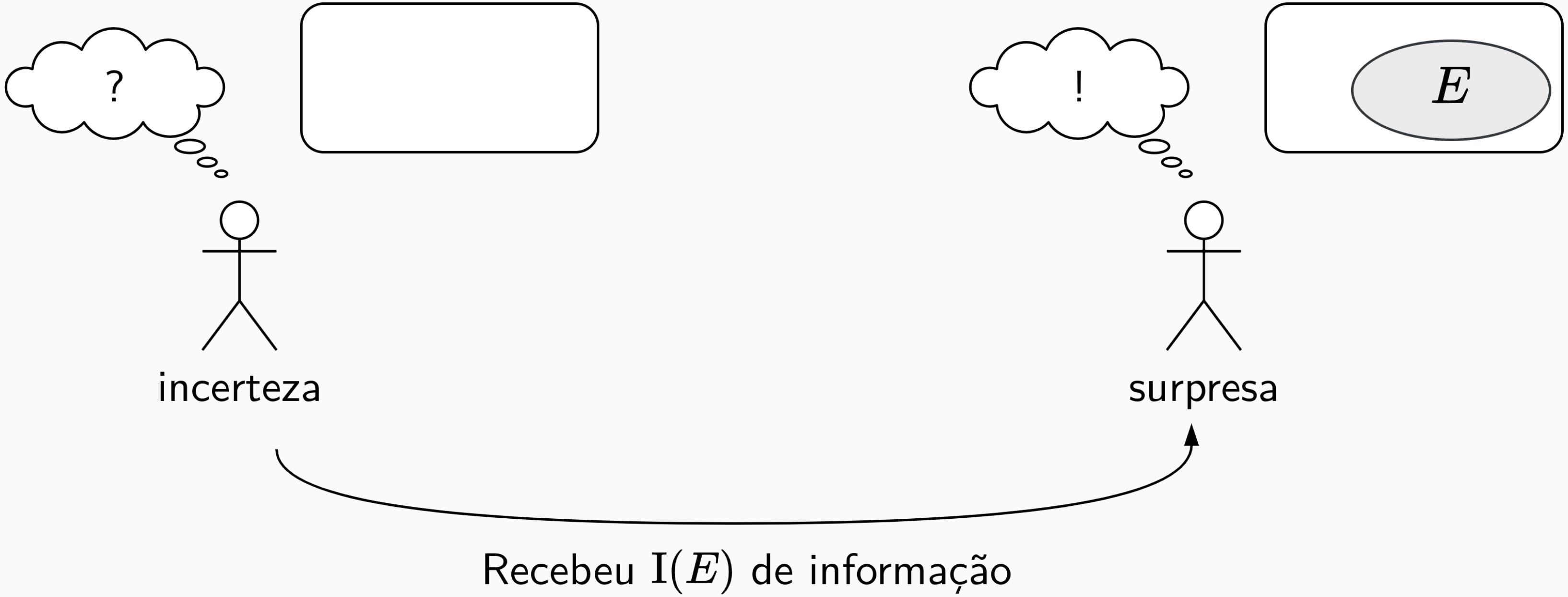
Conceito chave: **Capacidade do canal**.



Medida de informação

Como quantificar a informação?

Seja E um evento de um experimento probabilístico.



Como quantificar a informação recebida ao se observar E?

Axiomas da informação

1. Um evento que ocorre com probabilidade 1 não fornece informação.
   Se $\mathbb{P}[E] = 1$, então $\mathrm{I}(E) = 0$.

Axiomas da informação

1. Um evento que ocorre com probabilidade 1 não fornece informação.
   Se $\mathbb{P}[E] = 1$, então $\mathrm{I}(E) = 0$.

2. Quanto mais improvável um evento, maior a informação recebida ao observá-lo.
   Se $\mathbb{P}[E_1] < \mathbb{P}[E_2]$, então $\mathrm{I}(E_1) > \mathrm{I}(E_2)$.

Axiomas da informação

1. Um evento que ocorre com probabilidade 1 não fornece informação.
   Se $\mathbb{P}[E] = 1$, então $\mathrm{I}(E) = 0$.

2. Quanto mais improvável um evento, maior a informação recebida ao observá-lo.
   Se $\mathbb{P}[E_1] < \mathbb{P}[E_2]$, então $\mathrm{I}(E_1) > \mathrm{I}(E_2)$.

3. A informação recebida ao observar dois eventos independentes é a soma das suas informações individuais.
   Se $\mathbb{P}[E_1 \cap E_2] = \mathbb{P}[E_1]\mathbb{P}[E_2]$, então $\mathrm{I}(E_1 \cap E_2) = \mathrm{I}(E_1) + \mathrm{I}(E_2)$.

Axiomas da informação

1. Um evento que ocorre com probabilidade 1 não fornece informação.
   Se $\mathbb{P}[E] = 1$, então $\mathrm{I}(E) = 0$.

2. Quanto mais improvável um evento, maior a informação recebida ao observá-lo.
   Se $\mathbb{P}[E_1] < \mathbb{P}[E_2]$, então $\mathrm{I}(E_1) > \mathrm{I}(E_2)$.

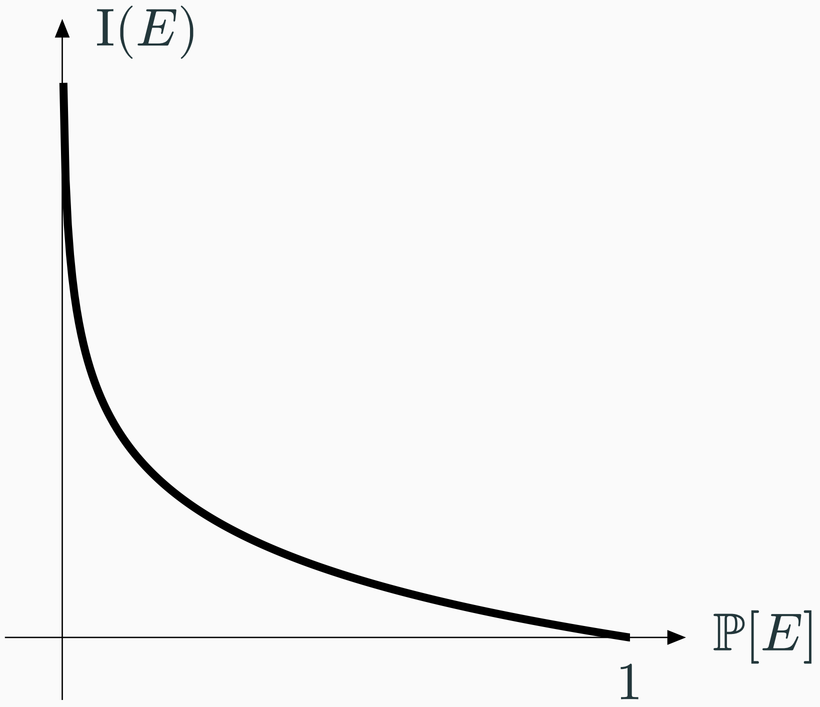
3. A informação recebida ao observar dois eventos independentes é a soma das suas informações individuais.
   Se $\mathbb{P}[E_1 \cap E_2] = \mathbb{P}[E_1]\mathbb{P}[E_2]$, então $\mathrm{I}(E_1 \cap E_2) = \mathrm{I}(E_1) + \mathrm{I}(E_2)$.

A função I que satisfaz os três axiomas é **única**, a menos de uma constante de escala.

Informação de Shannon

Medida de informação proposta por Shannon:

$$\mathrm{I}(E) = \log \frac{1}{\mathbb{P}[E]}$$



Unidades de informação

A base do logaritmo equivale à constante de escala e determina a *unidade* da informação.

Base	**Unidade**
2	bit ou shannon
3	trit
$e = 2.7183...$	nat
10	dígito decimal ou hartley

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

   $\mathrm{I}(\texttt{cara}) = \log 2 = 1 \text{ bit}, \qquad \mathrm{I}(\texttt{coroa}) = \log 2 = 1 \text{ bit}.$

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

   $\mathrm{I}(\mathtt{cara}) = \log 2 = 1\ \text{bit}, \qquad \mathrm{I}(\mathtt{coroa}) = \log 2 = 1\ \text{bit}.$

2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\mathtt{cara}] = \frac{1}{4}$? E coroa?

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?
   $\mathrm{I}(\mathsf{cara}) = \log 2 = 1 \text{ bit}, \qquad \mathrm{I}(\mathsf{coroa}) = \log 2 = 1 \text{ bit}.$
2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\mathsf{cara}] = \frac{1}{4}$? E coroa? $\mathrm{I}(\mathsf{cara}) = \log 4 = 2 \text{ bits}, \qquad \mathrm{I}(\mathsf{coroa}) = \log(4/3) = 0.42 \text{ bits}.$

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?
   $\mathrm{I}(\mathsf{cara}) = \log 2 = 1 \text{ bit}, \qquad \mathrm{I}(\mathsf{coroa}) = \log 2 = 1 \text{ bit}.$
2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\mathsf{cara}] = \frac{1}{4}$? E coroa? $\mathrm{I}(\mathsf{cara}) = \log 4 = 2 \text{ bits}, \qquad \mathrm{I}(\mathsf{coroa}) = \log(4/3) = 0.42 \text{ bits}.$
3. Caiu ⚄ após o lançamento de um dado honesto?

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?
   $\mathrm{I}(\mathtt{cara}) = \log 2 = 1$ bit, $\quad \mathrm{I}(\mathtt{coroa}) = \log 2 = 1$ bit.
2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\mathtt{cara}] = \frac{1}{4}$? E coroa? $\mathrm{I}(\mathtt{cara}) = \log 4 = 2$ bits, $\quad \mathrm{I}(\mathtt{coroa}) = \log(4/3) = 0.42$ bits.
3. Caiu ⚄ após o lançamento de um dado honesto?
   $\mathrm{I}(⚄) = \log 6 = 2.58$ bits $= 1$ dígito senário.

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?
   $\mathrm{I}(\texttt{cara}) = \log 2 = 1 \text{ bit}, \qquad \mathrm{I}(\texttt{coroa}) = \log 2 = 1 \text{ bit}.$
2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\texttt{cara}] = \frac{1}{4}$? E coroa? $\mathrm{I}(\texttt{cara}) = \log 4 = 2 \text{ bits}, \qquad \mathrm{I}(\texttt{coroa}) = \log(4/3) = 0.42 \text{ bits}.$
3. Caiu ⚄ após o lançamento de um dado honesto?
   $\mathrm{I}(\text{⚄}) = \log 6 = 2.58 \text{ bits} = 1 \text{ dígito senário}.$
4. Está 🟡 em um semáforo com $\mathbb{P}[\text{🔴}] = 0.80, \mathbb{P}[\text{🟡}] = 0.01, \mathbb{P}[\text{🟢}] = 0.19$? E 🔴? E 🟢?

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?
   $\mathrm{I}(\texttt{cara}) = \log 2 = 1 \text{ bit}, \qquad \mathrm{I}(\texttt{coroa}) = \log 2 = 1 \text{ bit}.$
2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\texttt{cara}] = \frac{1}{4}$? E coroa? $\mathrm{I}(\texttt{cara}) = \log 4 = 2 \text{ bits}, \qquad \mathrm{I}(\texttt{coroa}) = \log(4/3) = 0.42 \text{ bits}.$
3. Caiu ⚄ após o lançamento de um dado honesto?
   $\mathrm{I}(⚄) = \log 6 = 2.58 \text{ bits} = 1 \text{ dígito senário}.$
4. Está 🟡 em um semáforo com $\mathbb{P}[$🔴$] = 0.80$, $\mathbb{P}[$🟡$] = 0.01$, $\mathbb{P}[$🟢$] = 0.19$? E 🔴? E 🟢?
   $\mathrm{I}($🟡$) = \log(1/0.01) = 6.64 \text{ bits}.\qquad \mathrm{I}($🔴$) = \log(1/0.80) = 0.32 \text{ bits}.\qquad \mathrm{I}($🟢$) = \log(1/0.19) = 2.40 \text{ bits}.$

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

   $\mathrm{I}(\text{cara}) = \log 2 = 1 \text{ bit}, \qquad \mathrm{I}(\text{coroa}) = \log 2 = 1 \text{ bit}.$

2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\text{cara}] = \frac{1}{4}$? E coroa? $\mathrm{I}(\text{cara}) = \log 4 = 2 \text{ bits}, \qquad \mathrm{I}(\text{coroa}) = \log(4/3) = 0.42 \text{ bits}.$

3. Caiu ⚄ após o lançamento de um dado honesto?

   $\mathrm{I}(⚄) = \log 6 = 2.58 \text{ bits} = 1 \text{ dígito senário}.$

4. Está 🟡 em um semáforo com $\mathbb{P}[$🔴$] = 0.80$, $\mathbb{P}[$🟡$] = 0.01$, $\mathbb{P}[$🟢$] = 0.19$? E 🔴? E 🟢?

   $\mathrm{I}($🟡$) = \log(1/0.01) = 6.64 \text{ bits}. \qquad \mathrm{I}($🔴$) = \log(1/0.80) = 0.32 \text{ bits}. \qquad \mathrm{I}($🟢$) = \log(1/0.19) = 2.40 \text{ bits}.$

5. Uma carta retirada do baralho é de ♠? Que é uma dama? Que é a dama de ♠?

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?
   $\mathrm{I}(\texttt{cara}) = \log 2 = 1 \text{ bit}, \quad \mathrm{I}(\texttt{coroa}) = \log 2 = 1 \text{ bit}.$
2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\texttt{cara}] = \frac{1}{4}$? E coroa? $\mathrm{I}(\texttt{cara}) = \log 4 = 2 \text{ bits}, \quad \mathrm{I}(\texttt{coroa}) = \log(4/3) = 0.42 \text{ bits}.$
3. Caiu ⚄ após o lançamento de um dado honesto?
   $\mathrm{I}(⚄) = \log 6 = 2.58 \text{ bits} = 1 \text{ dígito senário}.$
4. Está 🟡 em um semáforo com $\mathbb{P}[🔴] = 0.80, \mathbb{P}[🟡] = 0.01, \mathbb{P}[🟢] = 0.19$? E 🔴? E 🟢?
   $\mathrm{I}(🟡) = \log(1/0.01) = 6.64 \text{ bits}. \quad \mathrm{I}(🔴) = \log(1/0.80) = 0.32 \text{ bits}. \quad \mathrm{I}(🟢) = \log(1/0.19) = 2.40 \text{ bits}.$
5. Uma carta retirada do baralho é de ♠? Que é uma dama? Que é a dama de ♠?
   $\mathrm{I}(\spadesuit) = \log 4 = 2 \text{ bits}, \quad \mathrm{I}(\mathrm{Q}) = \log 13 = 3.7 \text{ bits}, \quad \mathrm{I}(\mathrm{Q}\spadesuit) = \log 13 = 5.7 \text{ bits}.$

Exemplos

6. Os quatro últimos dígitos do número de telefone do seu amigo é 5678?

Exemplos

6. Os quatro últimos dígitos do número de telefone do seu amigo é 5678?
   $\mathrm{I}(5678) = \log 10^4 = 13.28$ bits $= 4$ dígitos decimais.

Exemplos

6. Os quatro últimos dígitos do número de telefone do seu amigo é 5678?
   $\mathrm{I}(5678) = \log 10^4 = 13.28$ bits $= 4$ dígitos decimais.

7. O sol nasceu hoje?

Exemplos

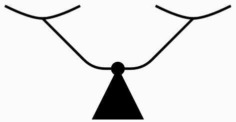
6. Os quatro últimos dígitos do número de telefone do seu amigo é 5678?
   $\mathrm{I}(5678) = \log 10^4 = 13.28$ bits $= 4$ dígitos decimais.
7. O sol nasceu hoje?
   Para Laplace, $\log(1/0.9999) = 0.00014$ bits.

Problema das 12 moedas

- Temos 12 moedas:
  1 2 3 4 5 6 7 8 9 10 11 12

- Sabemos que exatamente uma é falsa (pode ser ou mais leve ou mais pesada).

- Dispomos de uma balança de braços:



- Como descobrir qual é a moeda falsa (e se é mais leve ou mais pesada) com o menor número de pesagens?

Problema das 12 moedas

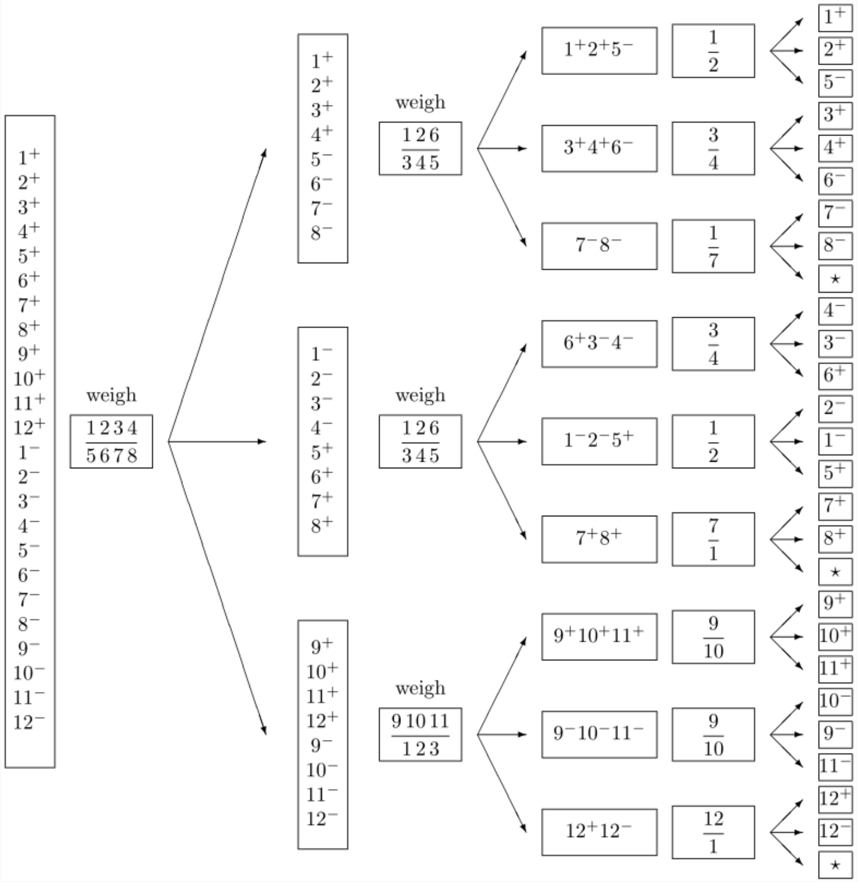
- Variável aleatória que queremos descobrir: $Y \in \{1^+, 1^-, 2^+, 2^-, ..., 12^+, 12^-\}$.
- Assumindo que Y é uniforme, temos:

  $$\mathrm{I}(Y = y) = \log 24 = 4.58 \text{ bits} = 2.89 \text{ trits}$$

  para todo y.
- Resultado da pesagem i: $X_i \in \{\mathsf{L}, \mathsf{B}, \mathsf{R}\}$.
- A cada pesagem i, ganhamos $\mathrm{I}(X_i = x_i)$ de informação.
- Heurística: maximizar o pior caso (menor valor possível de $\mathrm{I}(X_i = x_i)$).

Problema das 12 moedas

Fonte: MacKay [1].



Entropia

Entropia de uma variável aleatória

A entropia é a *quantidade de informação média* de uma variável aleatória.

Definição: Seja X uma va discreta com alfabeto $\mathcal{X}$ e pmf p. A **entropia** de X é dada por

$$\mathrm{H}(X) = \sum_{x \in \mathcal{X}} p(x) \mathrm{I}(X = x) = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)}.$$

Observação: Como a entropia depende apenas da distribuição p, utiliza-se também a notação $\mathrm{H}(X) = \mathrm{H}(p)$.

Exercício

Determine a entropia da variável aleatória X, considerando:

(a) $\mathcal{X} = \{$🔴, 🟡, 🟢$\}$ e $p = [0.80, 0.01, 0.19]$.

(b) $\mathcal{X} = \{$⚀, ⚁, ⚂, ⚃, ⚄, ⚅$\}$ e $p = \left[\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right]$.

Exercício

Determine a entropia da variável aleatória X, considerando:

(a) $\mathcal{X} = \{\textcolor{red}{\bullet}, \textcolor{yellow}{\bullet}, \textcolor{green}{\bullet}\}$ e $p = [0.80, 0.01, 0.19]$.

(b) $\mathcal{X} = \{\text{⚀}, \text{⚁}, \text{⚂}, \text{⚃}, \text{⚄}, \text{⚅}\}$ e $p = \left[\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right]$.

Resposta:

(a) 0.78 bits.

(b) $\log(6) = 2.58\text{bits} = 1$ dígito senário.

Entropia de uma Bernoulli

Seja $X \sim \mathrm{Bernoulli}(\alpha)$, isto é, $\mathcal{X} = \{\mathtt{0}, \mathtt{1}\}$ e $p = [1 - \alpha, \alpha]$.

Entropia de uma Bernoulli

Seja $X \sim \mathrm{Bernoulli}(\alpha)$, isto é, $\mathcal{X} = \{\mathtt{0}, \mathtt{1}\}$ e $p = [1-\alpha, \alpha]$.

- Por exemplo, $p = \left[\frac{1}{2}, \frac{1}{2}\right]$:

$$\mathrm{H}(X) = p(\mathtt{0}) \log \frac{1}{p(\mathtt{0})} + p(\mathtt{1}) \log \frac{1}{p(\mathtt{1})} = \frac{1}{2} \log 2 + \frac{1}{2} \log 2 = \log 2 = 1 \text{ bit}.$$

Entropia de uma Bernoulli

Seja $X \sim \text{Bernoulli}(\alpha)$, isto é, $\mathcal{X} = \{\mathtt{0}, \mathtt{1}\}$ e $p = [1-\alpha, \alpha]$.

- Por exemplo, $p = \left[\frac{1}{2}, \frac{1}{2}\right]$:

$$\mathrm{H}(X) = p(\mathtt{0}) \log \frac{1}{p(\mathtt{0})} + p(\mathtt{1}) \log \frac{1}{p(\mathtt{1})} = \frac{1}{2} \log 2 + \frac{1}{2} \log 2 = \log 2 = 1 \text{ bit}.$$

- Por exemplo, $p = \left[\frac{3}{4}, \frac{1}{4}\right]$:

$$\mathrm{H}(X) = p(\mathtt{0}) \log \frac{1}{p(\mathtt{0})} + p(\mathtt{1}) \log \frac{1}{p(\mathtt{1})} = \frac{1}{4} \log 4 + \frac{3}{4} \log \frac{4}{3} = 0.81 \text{ bits}.$$

Entropia de uma Bernoulli

Seja $X \sim \mathrm{Bernoulli}(\alpha)$, isto é, $\mathcal{X} = \{\mathtt{0}, \mathtt{1}\}$ e $p = [1-\alpha, \alpha]$.

- Por exemplo, $p = \left[\frac{1}{2}, \frac{1}{2}\right]$:

$$\mathrm{H}(X) = p(\mathtt{0}) \log \frac{1}{p(\mathtt{0})} + p(\mathtt{1}) \log \frac{1}{p(\mathtt{1})} = \frac{1}{2} \log 2 + \frac{1}{2} \log 2 = \log 2 = 1 \text{ bit}.$$

- Por exemplo, $p = \left[\frac{3}{4}, \frac{1}{4}\right]$:

$$\mathrm{H}(X) = p(\mathtt{0}) \log \frac{1}{p(\mathtt{0})} + p(\mathtt{1}) \log \frac{1}{p(\mathtt{1})} = \frac{1}{4} \log 4 + \frac{3}{4} \log \frac{4}{3} = 0.81 \text{ bits}.$$

- Por exemplo, $p = [0, 1]$:

$$\mathrm{H}(X) = p(\mathtt{0}) \log \frac{1}{p(\mathtt{0})} + p(\mathtt{1}) \log \frac{1}{p(\mathtt{1})} = 0 \log \frac{1}{0} + 1 \log 1 = 0 \text{ bits}.$$

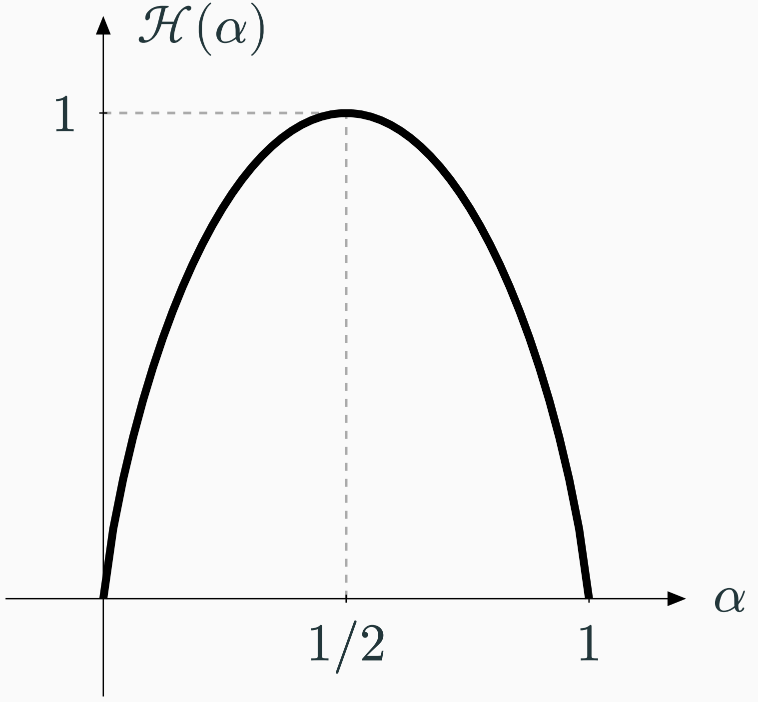
Entropia de uma Bernoulli

Seja $X \sim \text{Bernoulli}(\alpha)$, isto é, $\mathcal{X} = \{\texttt{0}, \texttt{1}\}$ e $p = [1-\alpha, \alpha]$.

Generalizando:

$$\begin{aligned} \mathrm{H}(X) &= p(\texttt{0}) \log \frac{1}{p(\texttt{0})} + p(\texttt{1}) \log \frac{1}{p(\texttt{1})} \\ &= (1-\alpha) \log \frac{1}{1-\alpha} + \alpha \log \frac{1}{\alpha} \stackrel{\text{def}}{=} \mathcal{H}(\alpha), \end{aligned}$$

onde $\mathcal{H}$ é chamada de **função entropia binária**.



Parênteses: $0 \log(1/0) = 0$

Fato:

$$\lim_{x \to 0} x \log \frac{1}{x} = 0.$$

Demonstração: Temos que

$$\begin{aligned} \lim_{x \to 0} x \log \frac{1}{x} &= \lim_{x \to 0} \frac{\log 1/x}{1/x} \\ &= \lim_{x \to 0} \frac{-1/x}{-1/x^2} \\ &= \lim_{x \to 0} x = 0, \end{aligned}$$

onde a segunda igualdade segue da regra de L'Hospital. ■

Propriedades da entropia

1. A entropia é uma função côncava de p.

2. A entropia satistaz

$$0 \overset{(a)}{\leq} \mathrm{H}(X) \overset{(b)}{\leq} \log|\mathcal{X}|,$$

com:

- Igualdade em (a) para X determinística.
- Igualdade em (b) para X uniforme sobre $\mathcal{X}$.

[desenhos da determinística e da uniforme]

Compressão de fonte sem perdas

Fontes de informação discretas

- Uma **fonte discreta** é qualquer sequência aleatória $\mathbf{X} = X_0 X_1 \cdots$, que assume valores em um alfabeto finito $\mathcal{X}$. Os elementos de $\mathcal{X}$ são chamados de **letras** ou **símbolos**.

Fontes de informação discretas

- Uma **fonte discreta** é qualquer sequência aleatória $\mathbf{X} = X_0 X_1 \cdots$, que assume valores em um alfabeto finito $\mathcal{X}$. Os elementos de $\mathcal{X}$ são chamados de **letras** ou **símbolos**.

- Uma fonte discreta $\mathbf{X} = X_0 X_1 \cdots$ é dita ser **sem memória** (DMS) quando $X_n \overset{\text{iid}}{\sim} p$ para alguma pmf p sobre $\mathcal{X}$. Nesse caso, removeremos o negrito da notação e escreveremos apenas $X = (\mathcal{X}, p)$ para representar a DMS.

Fontes de informação discretas

- Uma **fonte discreta** é qualquer sequência aleatória $\mathbf{X} = X_0 X_1 \cdots$, que assume valores em um alfabeto finito $\mathcal{X}$. Os elementos de $\mathcal{X}$ são chamados de **letras** ou **símbolos**.

- Uma fonte discreta $\mathbf{X} = X_0 X_1 \cdots$ é dita ser **sem memória** (DMS) quando $X_n \overset{\text{iid}}{\sim} p$ para alguma pmf p sobre $\mathcal{X}$. Nesse caso, removeremos o negrito da notação e escreveremos apenas $X = (\mathcal{X}, p)$ para representar a DMS.

Exemplo: Considere a DMS $X = (\mathcal{X}, p)$ com $\mathcal{X} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ e $p = [0.80, 0.15, 0.05]$. Uma possível sequência amostra dessa fonte poderia ser

$$\mathbf{x} = \mathsf{ababaaaacaaaaaaaaaacaaabaaccaaaaaaaaaaaaaabaaaaba}\ldots$$

Fontes de informação discretas

- Uma **fonte discreta** é qualquer sequência aleatória $\mathbf{X} = X_0 X_1 \cdots$, que assume valores em um alfabeto finito $\mathcal{X}$. Os elementos de $\mathcal{X}$ são chamados de **letras** ou **símbolos**.

- Uma fonte discreta $\mathbf{X} = X_0 X_1 \cdots$ é dita ser **sem memória** (DMS) quando $X_n \overset{\text{iid}}{\sim} p$ para alguma pmf p sobre $\mathcal{X}$. Nesse caso, removeremos o negrito da notação e escreveremos apenas $X = (\mathcal{X}, p)$ para representar a DMS.

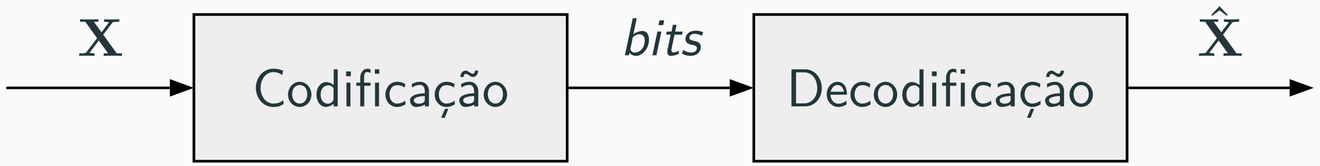
Exemplo: Considere a DMS $X = (\mathcal{X}, p)$ com $\mathcal{X} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ e $p = [0.80, 0.15, 0.05]$. Uma possível sequência amostra dessa fonte poderia ser

$$\mathbf{x} = \mathtt{ababaaaaacaaaaaaaaaacaaabaaccaaaaaaaaaaaaaabaaaaba...}$$

Observação: Há outros modelos de fontes discretas. Um modelo mais geral, descrito por Shannon em seu artigo seminal [2], consiste na *fonte discreta markoviana*. No entanto, nesta disciplina, estudaresmos apenas as fontes discretas sem memória.

Compressão de fonte

Objetivo: Representar com *fidelidade* uma fonte de informação através de uma sequência de bits com a *menor taxa* possível.



- Compressão **sem perdas** (*lossless*): $\hat{\mathbf{X}} = \mathbf{X}$.
- Compressão **com perdas** (*lossy*): $\hat{\mathbf{X}} \approx \mathbf{X}$.

  Exemplo: minimizar $\mathbb{P}\big[\hat{\mathbf{X}} \neq \mathbf{X}\big]$ ou alguma *função de distorção* $d\big(\hat{\mathbf{X}}, \mathbf{X}\big)$.

Nesta disciplina: Apenas os fundamentos de compressão sem perdas de fontes discretas.

Notação

Seja $\mathcal{X}$ um conjunto. Define-se:

- **Potência cartesiana**: $\mathcal{X}^n = \{x_1 x_2 \cdots x_n \mid x_i \in \mathcal{X}\}$.
  Ou seja, $\mathcal{X}^n$ é o conjunto de todas as sequências com elementos em $\mathcal{X}$ de comprimento exatamente n.
  Exemplo: $\{\mathsf{a}, \mathsf{b}, \mathsf{c}\}^2 = \{\mathsf{aa}, \mathsf{ab}, \mathsf{ac}, \mathsf{ba}, \mathsf{bb}, \mathsf{bc}, \mathsf{ca}, \mathsf{cb}, \mathsf{cc}\}$.

- **Kleene star**: $\mathcal{X}^* = \mathcal{X}^0 \cup \mathcal{X}^1 \cup \mathcal{X}^2 \cup \cdots$.
  Ou seja, $\mathcal{X}^*$ é o conjunto de todas as sequências finitas com elementos em $\mathcal{X}$.
  Exemplo: $\{\mathsf{a}, \mathsf{b}, \mathsf{c}\}^* = \{\varepsilon, \mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{aa}, \mathsf{ab}, \mathsf{ac}, \mathsf{ba}, \mathsf{bb}, \mathsf{bc}, \mathsf{ca}, \mathsf{cb}, \mathsf{cc}, \mathsf{aaa}, \mathsf{aab}, \ldots\}$

- **Kleene plus**: $\mathcal{X}^+ = \mathcal{X}^1 \cup \mathcal{X}^2 \cup \cdots$.
  Ou seja, $\mathcal{X}^+$ é o conjunto de todas as sequências finitas não-vazias com elementos em $\mathcal{X}$.
  Exemplo: $\{\mathsf{a}, \mathsf{b}, \mathsf{c}\}^+ = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{aa}, \mathsf{ab}, \mathsf{ac}, \mathsf{ba}, \mathsf{bb}, \mathsf{bc}, \mathsf{ca}, \mathsf{cb}, \mathsf{cc}, \mathsf{aaa}, \mathsf{aab}, \ldots\}$

Notação: A **sequência vazia** é denotada por ε.

Definição: Um **código de fonte símbolo-a-símbolo** para uma fonte discreta (qualquer) com alfabeto $\mathcal{X}$ é dado por um mapeamento

$$C : \mathcal{X} \to \{\mathtt{0}, \mathtt{1}\}^+.$$

Exemplo: Os quatro códigos abaixo são todos para uma fonte com $\mathcal{X} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$.

x	$C_1(x)$	$C_2(x)$	$C_3(x)$	$C_4(x)$
a	00	1	0	0
b	01	01	01	1
c	10	000	011	00
d	11	001	0111	11

Os elementos da imagem de C são chamados de **palavras-código**.

Extensão de um código símbolo-a-símbolo

Estende-se o código para qualquer sequência $\mathbf{x} \in \mathcal{X}^+$ *concatenativamente*. Abusando da notação, também chamaremos essa extensão de $C : \mathcal{X}^+ \to \{\mathtt{0}, \mathtt{1}\}^+$.

Exemplo: Considere novamente os quatro códigos abaixo.

x	$C_1(x)$	$C_2(x)$	$C_3(x)$	$C_4(x)$
a	00	1	0	0
b	01	01	01	1
c	10	000	011	00
d	11	001	0111	11

Por exemplo, para $\mathbf{x} = \mathtt{babaca}$:

- $C_1(\mathbf{x}) = \mathtt{010001001000}$.
- $C_2(\mathbf{x}) = \mathtt{0110110001}$.
- $C_3(\mathbf{x}) = \mathtt{0100100110}$.
- $C_4(\mathbf{x}) = \mathtt{1010000}$.

Códigos unicamente decodificáveis

- Um código é dito ser **unicamente decodificável** se o mapeamento estendido for injetivo:

$$\mathbf{x} \neq \mathbf{x}' \implies C(\mathbf{x}) \neq C(\mathbf{x}'),$$

  para todos $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^+$.

- No exemplo:
  - C_1, C_2 e C_3 são unicamente decodificáveis.
  - C_4 não é unicamente decodificável.
    Por exemplo, $C_4(\mathsf{aa}) = \texttt{00} = C_4(\mathsf{c})$.

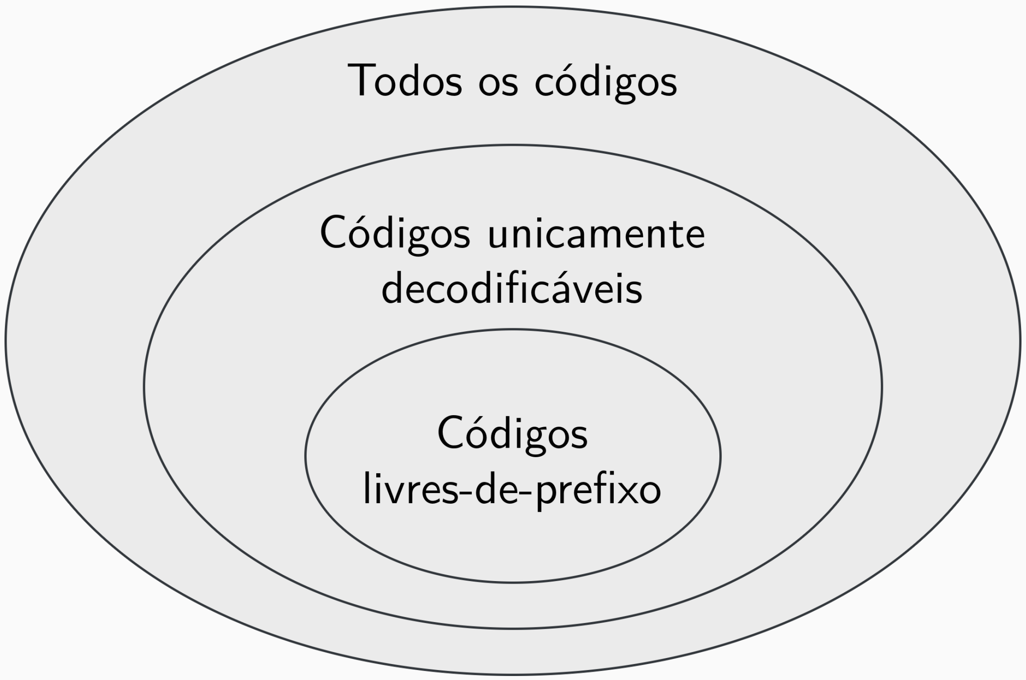
- Há diversos algoritmos para verificar se um dado código é unicamente decodificável. Talvez o mais famoso deles seja o *algoritmo de Sardinas–Patterson (1953)*.

Códigos livres-de-prefixo

- Um código é dito ser **livre-de-prefixo** (ou **instantâneo**) se nenhuma palavra-código é prefixo de outra.
- No exemplo:
  - C_1 e C_2 são livres-de-prefixo.
  - C_3 e C_4 não são livres-de-prefixo.
- Verificar se um código é livre-de-prefixo é trivial.
- Decodificar um código livre-de-prefixo também é trivial (e "instantâneo").

Diagrama de Venn das classes de códigos

Fato: Livre-de-prefixo $\Longrightarrow$ unicamente decodificável.



Teoremas de codificação de fonte

Desigualdade de Kraft–McMillan

Teorema. Seja $\ell(x)$ o número de bits em $C(x)$, para $x \in \mathcal{X}$. Então:

1. *(B. McMillan, 1956)* Todo código unicamente decodificável satisfaz

$$\sum_{x \in \mathcal{X}} \frac{1}{2^{\ell(x)}} \leq 1. \qquad (*)$$

2. *(L. G. Kraft, 1949)* Dados comprimentos $\ell(x) : x \in \mathcal{X}$ que satisfazem (*), é possível construir um código livre-de-prefixo com tais comprimentos.

Verdadeiro ou falso?

(a) Não existe um código com comprimentos $1, 2, 3, 3, 4$ que seja unicamente decodificável.

(b) Não existe um código com comprimentos $1, 2, 3, 3, 4$ que seja livre de prefixo.

(c) Todo código com comprimentos $2, 2, 3, 3, 4, 4, 5$ é unicamente de codificável.

(d) Existe um código com comprimentos $2, 2, 3, 3, 4, 4, 5$ que seja livre-de-prefixo.

(e) Existe um código com comprimentos $2, 2, 3, 3, 4, 4, 5$ que seja unicamente decodificável.

Exercício

Verdadeiro ou falso?

(a) Não existe um código com comprimentos $1, 2, 3, 3, 4$ que seja unicamente decodificável.

(b) Não existe um código com comprimentos $1, 2, 3, 3, 4$ que seja livre de prefixo.

(c) Todo código com comprimentos $2, 2, 3, 3, 4, 4, 5$ é unicamente de codificável.

(d) Existe um código com comprimentos $2, 2, 3, 3, 4, 4, 5$ que seja livre-de-prefixo.

(e) Existe um código com comprimentos $2, 2, 3, 3, 4, 4, 5$ que seja unicamente decodificável.

Resposta:

(a) V. (b) V. (c) F. (d) V. (e) V.

0	00	000	0000
			0001
		001	0010
			0011
	01	010	0100
			0101
		011	0110
			0111
1	10	100	1000
			1001
		101	1010
			1011
	11	110	1100
			1101
		111	1110
			1111

Fonte: MacKay [1].

Comprimento médio de um código

O **comprimento médio** de um código C em uma DMS $X = (\mathcal{X}, p)$ é definido por

$$\bar{\ell}(C, X) = \mathbb{E}[\ell(X)] = \sum_{x \in \mathcal{X}} p(x)\ell(x).$$

onde $\ell(x)$ é o número de bits em $C(x)$, para $x \in \mathcal{X}$.

Exercício

Seja $\mathcal{X} = \{a, b, c, d\}$. Considere as seguintes fontes discretas sem memória:

- $X_1 = (\mathcal{X}, p_1)$, com $p_1 = [0.6, 0.2, 0.1, 0.1]$ e
- $X_2 = (\mathcal{X}, p_2)$, com $p_2 = [0.3, 0.3, 0.2, 0.2]$.

Determine o comprimento médio de cada código abaixo em cada DMS.

(a) $C_1 = [\texttt{00}, \texttt{01}, \texttt{10}, \texttt{11}]$.

(b) $C_2 = [\texttt{0}, \texttt{10}, \texttt{110}, \texttt{111}]$.

Exercício

Seja $\mathcal{X} = \{a, b, c, d\}$. Considere as seguintes fontes discretas sem memória:

- $X_1 = (\mathcal{X}, p_1)$, com $p_1 = [0.6, 0.2, 0.1, 0.1]$ e
- $X_2 = (\mathcal{X}, p_2)$, com $p_2 = [0.3, 0.3, 0.2, 0.2]$.

Determine o comprimento médio de cada código abaixo em cada DMS.

(a) $C_1 = [\texttt{00}, \texttt{01}, \texttt{10}, \texttt{11}]$.

(b) $C_2 = [\texttt{0}, \texttt{10}, \texttt{110}, \texttt{111}]$.

Resposta:

(a) $\bar{\ell}(C_1, X_1) = 2.0 \text{ bits/letra}$ e $\bar{\ell}(C_1, X_2) = 2.0 \text{ bits/letra}$.

(b) $\bar{\ell}(C_2, X_1) = 1.6 \text{ bits/letra}$ e $\bar{\ell}(C_2, X_2) = 2.1 \text{ bits/letra}$.

Exemplo

Exemplo: DMS $X = (\mathcal{X}, p)$ com $\mathcal{X} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ e $p = \left[\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right]$.

x	$C_1(x)$	$C_2(x)$	$C_3(x)$	$C_4(x)$
a	00	1	0	0
b	01	01	01	1
c	10	000	011	00
d	11	001	0111	11

$$\bar{\ell}(C_1, X) = \tfrac{1}{2}(2) + \tfrac{1}{4}(2) + \tfrac{1}{8}(2) + \tfrac{1}{8}(2) = 2 \text{ bits/letra.}$$

$$\bar{\ell}(C_2, X) = \tfrac{1}{2}(1) + \tfrac{1}{4}(2) + \tfrac{1}{8}(3) + \tfrac{1}{8}(3) = 1.75 \text{ bits/letra.}$$

$$\bar{\ell}(C_3, X) = \tfrac{1}{2}(1) + \tfrac{1}{4}(2) + \tfrac{1}{8}(3) + \tfrac{1}{8}(4) = 1.875 \text{ bits/letra.}$$

$$\bar{\ell}(C_4, X) = \tfrac{1}{2}(1) + \tfrac{1}{4}(1) + \tfrac{1}{8}(2) + \tfrac{1}{8}(2) = 1.25 \text{ bits/letra.}$$

Teorema da codificação de fonte sem perdas de Shannon

Teorema. *(C. E. Shannon, 1948)* Seja $X = (\mathcal{X}, p)$ uma DMS. Então:

1. Todo código unicamente decodificável tem comprimento médio $\bar{\ell}(C, X) \geq \mathrm{H}(X)$.
2. Existe um código livre-de-prefixo com comprimento médio $\bar{\ell}(C, X) < \mathrm{H}(X) + 1$.

Observação: A entropia de uma DMS é definida pela entropia da pmf correspondente, isto é, $\mathrm{H}(X) = \mathrm{H}(p)$.

O problema do "bit extra" do teorema de Shannon

Exemplo: Considere a DMS $X = (\mathcal{X}, p)$ com $\mathcal{X} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ e $p = [0.98, 0.01, 0.01]$.

A entropia da fonte é

$$\mathrm{H}(X) = 0.98 \log\left(\tfrac{1}{0.98}\right) + 0.01 \log\left(\tfrac{1}{0.01}\right) + 0.01 \log\left(\tfrac{1}{0.01}\right) = 0.161 \text{ bits/letra.}$$

Já o código ótimo ($\mathsf{a} \mapsto \mathtt{0}$, $\mathsf{b} \mapsto \mathtt{10}$, $\mathsf{c} \mapsto \mathtt{11}$) tem comprimento médio

$$\bar{\ell}(C, X) = 0.98(1) + 0.01(2) + 0.01(2) = 1.02 \text{ bits/letra.}$$

- Comprimir 1000 letras da fonte com o código ótimo resulta em 1020 bits (em média).
- Mas temos esperanças de chegar em 161 bits (em média)!

Extensão de uma DMS

Ideia: Agrupar as letras emitidas pela fonte em "superletras" (bloco de k letras).

$$x^{(k)} = (x_1, x_2, ..., x_k).$$

Definição. Seja $X = (\mathcal{X}, p)$ uma DMS. A **extensão de k-ésima ordem** de X, denotada por $X^{(k)}$, tem alfabeto $\mathcal{X}^k$ e pmf dada por

$$p^{(k)}(x_1, x_2, ..., x_k) = \prod_{i=1}^{k} p(x_i).$$

Observação: Um código para $X^{(k)}$ de comprimento médio $\bar{\ell}$ equivale a um código para X de comprimento médio $\bar{\ell}/k$.

Exercício

Considere a DMS $X = (\mathcal{X}, p)$ com $\mathcal{X} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ e $p = [0.98, 0.01, 0.01]$.

(a) Determine a entropia de X.

(b) Determine $X^{(2)}$, isto é, a extensão de segunda ordem de X.

(c) Determine a entropia de X, em bits/superletra.

Exercício

Considere a DMS $X = (\mathcal{X}, p)$ com $\mathcal{X} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ e $p = [0.98, 0.01, 0.01]$.

(a) Determine a entropia de X.

(b) Determine $X^{(2)}$, isto é, a extensão de segunda ordem de X.

(c) Determine a entropia de X, em bits/superletra.

Resposta:

(a) $\mathrm{H}(X) = 0.161$ bits/letra.

(b)

$x^{(2)}$	aa	ab	ac	ba	bb	bc	ca	cb	cc
$p^{(2)}(x^{(2)})$	0.9604	0.0098	0.0098	0.0098	0.0001	0.0001	0.0098	0.0001	0.0001

(c) $\mathrm{H}(X^{(2)}) = 0.322$ bits/superletra.

“Solução” do problema do bit extra

De fato, é possivel provar que

$$\mathrm{H}(X^{(k)}) = k\mathrm{H}(X).$$

Aplicando o teorema de Shannon à fonte estendida:

$$\mathrm{H}(X^{(k)}) \overset{(\forall)}{\leq} \bar{\ell} \overset{(\exists)}{<} \mathrm{H}(X^{(k)}) + 1.$$

Substituindo $\mathrm{H}(X^{(k)}) = k\mathrm{H}(X)$ e dividindo por k:

$$\mathrm{H}(X) \overset{(\forall)}{\leq} \frac{\bar{\ell}}{k} \overset{(\exists)}{<} \mathrm{H}(X) + \frac{1}{k}.$$

No exemplo, para $k = 100$, o limite inferior continua $\mathrm{H}(X) = 0.161$ bits/letra, mas o limite superior diminui para $0.161 + 0.001 = 0.162$.

“Solução” do problema do bit extra

De fato, é possivel provar que

$$\mathrm{H}(X^{(k)}) = k\mathrm{H}(X).$$

Aplicando o teorema de Shannon à fonte estendida:

$$\mathrm{H}(X^{(k)}) \overset{(\forall)}{\leq} \bar{\ell} \overset{(\exists)}{<} \mathrm{H}(X^{(k)}) + 1.$$

Substituindo $\mathrm{H}(X^{(k)}) = k\mathrm{H}(X)$ e dividindo por k:

$$\mathrm{H}(X) \overset{(\forall)}{\leq} \frac{\bar{\ell}}{k} \overset{(\exists)}{<} \mathrm{H}(X) + \frac{1}{k}.$$

Observação: O problema do conceito de extensão de fonte é a *complexidade computacional*, que cresce exponencialmente com k.

Código de Huffman

Código de Huffman

São códigos símbolo-a-símbolo **ótimos** no sentido de minimizar o comprimento médio. São sempre livres-de-prefixo.

O primeiro passo é ordenar as probabilidades em *ordem decrescente*.

Exemplo 1

DMS $X = (\mathcal{X}, p)$ com $\mathcal{X} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}$ e $p_X = \left[\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right]$, com $\mathrm{H}(X) = 1.75$ bits/letra.

$$\bar{\ell} = \frac{1}{2}(1) + \frac{1}{4}(2) + \frac{1}{8}(3) + \frac{1}{8}(3) = 1.75 \text{ bits/letra}.$$

Exemplo 2

DMS com $\mathcal{X} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}\}$ e $p_X = [0.4, 0.2, 0.2, 0.1, 0.1]$, com $\mathrm{H}(X) = 2.122$ bits/letra.

$$\bar{\ell} = 0.4(2) + 0.2(2) + 0.2(2) + 0.1(3) + 0.1(3) = 2.2 \text{ bits/letra.}$$

Códigos de Lempel–Ziv

Códigos de Lempel–Ziv

Os **códigos de Lempel–Ziv** não necessitam do modelo probabilístico da fonte, nem assumem fonte sem memória (DMS). De fato, são *assintoticamente ótimos* para *fontes ergódicas*.

Ideia: Substituir subsequências por *referências* a ocorrências prévias da mesma subsequência.

- **LZ77** *(Ziv & Lempel, 1977)*: Janela deslizante.
- **LZ78** *(Ziv & Lempel, 1978)*: Dicionário explícito.

Ambos os esquemas têm como entrada uma sequência de letras de um alfabeto $\mathcal{X}$ e saída uma sequência de letras de um alfabeto $\mathcal{Y}$.

Por simplicidade, assumiremos alfabeto de saída binário: $\mathcal{Y} = \{0, 1\}$. A generalização para $\mathcal{Y}$ qualquer é imediata.

Código de Lempel–Ziv 77

Considera uma **janela deslizante** sobre a mensagem. Esta janela é dividida em 2 partes:

- **Search buffer**, de comprimento S, e
- **Lookahead buffer**, de comprimento L.

Observação: O tamanho total da janela é $W = S + L$.

A cada passo do algoritmo, um token no formato (p, ℓ, x) é emitido, onde:

- $p \in [0 : S)$ é o **ponteiro** do match,
- $\ell \in [0 : L)$ é o **comprimento** do match, e
- $x \in \mathcal{X}$ é a **inovação**, que é letra da fonte após o match.

Ao final de cada passo, avança-se $\ell + 1$ posições.

Observação: O conteúdo inicial do search buffer é arbitrário; aqui, consideraremos "zerado".

Exemplo LZ77: Codificação

- *Alfabeto de entrada*: $\mathcal{X} = \{\text{␣}, \mathsf{a}, \mathsf{b}, \mathsf{c}, \ldots, \mathsf{z}\}$, com $|\mathcal{X}| = 27$.
- *Parâmetros da janela:* comprimento total $W = 12$, com $S = 8$ e $L = 4$.
- *Mensagem:* $\mathbf{x} =$ a␣asa␣da␣casa.

  Não comprimido: 65 bits.

- *Tokens:* $(7, 0, \mathsf{a})$, $(6, 2, \mathsf{s})$, $(4, 2, \mathsf{d})$, $(5, 2, \mathsf{c})$, $(0, 2, \mathsf{a})$.

Exemplo LZ77: Decodificação

- *Tokens:* $(7, 0, \mathsf{a})$, $(6, 2, \mathsf{s})$, $(4, 2, \mathsf{d})$, $(5, 2, \mathsf{c})$, $(0, 2, \mathsf{a})$.

- *Mensagem recuperada*: $\hat{\mathbf{x}} =$ a␣asa␣da␣casa.

LZ77: Conversão tokens $\leftrightarrow$ bits

Cada token (p, ℓ, x) deve ser convertido em uma sequência de bits:

- $p \in [0 : S)$: requer $\lceil \log_2 S \rceil$ bits.
  No exemplo, $S = 8$, de modo que p requer 3 bits.
- $\ell \in [0 : L)$: requer $\lceil \log_2 L \rceil$ bits.
  No exemplo, $L = 4$, de modo que ℓ requer 2 bits.
- $x \in \mathcal{X}$: requer $\lceil \log_2 |\mathcal{X}| \rceil$ bits.
  No exemplo, $|\mathcal{X}| = 27$, de modo que x requer 5 bits.

Exemplo LZ77: tokens $\leftrightarrow$ bits

- *Tokens:* $(7, 0, \mathsf{a})$, $(6, 2, \mathsf{s})$, $(4, 2, \mathsf{d})$, $(5, 2, \mathsf{c})$, $(0, 2, \mathsf{a})$.

token (p, ℓ, x)	bits
$(7, 0, \mathsf{a})$	$(\mathtt{111}, \mathtt{00}, \mathtt{00001})$
$(6, 2, \mathsf{s})$	$(\mathtt{110}, \mathtt{10}, \mathtt{10011})$
$(4, 2, \mathsf{d})$	$(\mathtt{100}, \mathtt{10}, \mathtt{00100})$
$(5, 2, \mathsf{c})$	$(\mathtt{101}, \mathtt{10}, \mathtt{00011})$
$(0, 2, \mathsf{a})$	$(\mathtt{000}, \mathtt{10}, \mathtt{00001})$

- *Bits:* $\mathbf{y} =$ `111 00 00001 110 10 10011 100 10 00100 101 10 00011 000 10 00001`.
  Comprimido: 50 bits.

Código de Lempel–Ziv 78

Neste algoritmo:

- A mensagem é segmentada em *subsequências que ainda não ocorreram*.
- Um **dicionário** é construído à medida em que os segmentos são processados.

A cada passo do algoritmo, um token no formato (p, x) é emitido, onde:

- $p \in \mathbb{N}$ é o **ponteiro** (índice) do dicionário.
- $x \in \mathcal{X}$ é a **inovação**, que é letra da fonte após o match no dicionário.

Observação: O dicionário inicia com $\{0 : \varepsilon\}$.

Exemplo LZ78: Codificação

- *Alfabeto de entrada*: $\mathcal{X} = \{␣, \mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots, \mathtt{z}\}$, com $|\mathcal{X}| = 27$.
- *Mensagem:* $\mathbf{x} =$ `a␣asa␣da␣casa`.

  Não comprimido: 65 bits.

- *Tokens:* $(0, \mathtt{a})$, $(0, ␣)$, $(1, \mathtt{s})$, $(1, ␣)$, $(0, \mathtt{d})$, $(4, \mathtt{c})$, $(3, \mathtt{a})$.

Exemplo LZ78: Decodificação

- *Tokens:* $(0, \mathsf{a})$, $(0, \text{␣})$, $(1, \mathsf{s})$, $(1, \text{␣})$, $(0, \mathsf{d})$, $(4, \mathsf{c})$, $(3, \mathsf{a})$.

- *Mensagem recuperada*: $\hat{\mathbf{x}} = $ a␣asa␣da␣casa.

LZ78: Conversão tokens $\leftrightarrow$ bits

Cada token (p, x) deve ser convertido em uma sequência de bits:

- A inovação $x \in \mathcal{X}$ requer $\lceil \log_2 |\mathcal{X}| \rceil$ bits.
  No exemplo, $|\mathcal{X}| = 27$, de modo que x requer 5 bits.
- O ponteiro $p \in \mathbb{N}$ poderia ser representado com um número fixo de $\lceil \log_2 N \rceil$ bits, onde N é o número total de tokens.
  No exemplo, $N = 7$, de modo que p iria requerer 3 bits.
- No entanto, há uma maneira alternativa: como o ponteiro p do passo i só assume valores em $[0 : i)$, é possível representá-lo com $\lceil \log_2 i \rceil$ bits.
  Isso economiza alguns bits às custas de um tamanho variável da representação de cada token.

Exemplo LZ78: tokens $\leftrightarrow$ bits

- *Tokens:* $(0, \mathsf{a})$, $(0, \textvisiblespace)$, $(1, \mathsf{s})$, $(1, \textvisiblespace)$, $(0, \mathsf{d})$, $(4, \mathsf{c})$, $(3, \mathsf{a})$,

i	$\lceil \log_2 i \rceil$	token (p, ℓ, x)	bits
1	0	$(0, \mathsf{a})$	$(, \texttt{00001})$
2	1	$(0, \textvisiblespace)$	$(\texttt{0}, \texttt{00000})$
3	2	$(1, \mathsf{s})$	$(\texttt{01}, \texttt{10011})$
4	2	$(1, \textvisiblespace)$	$(\texttt{01}, \texttt{00000})$
5	3	$(0, \mathsf{d})$	$(\texttt{000}, \texttt{00100})$
6	3	$(4, \mathsf{c})$	$(\texttt{100}, \texttt{00011})$
7	3	$(3, \mathsf{a})$	$(\texttt{011}, \texttt{00001})$

- *Bits*: $\mathbf{y} =$ `00001 0 00000 01 10011 01 00000 000 00100 100 00011 011 00001`.
  Comprimido: 49 bits.

Referências

Referências

[1] David J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[2] C. E. Shannon e W. Weaver, *The Mathematical Theory of Communication*. University of Illinois Press, 1949.

[3] T. M. Cover e J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 2006.