

LEONARDO ARAÚJO

# TEORIA DA INFORMAÇÃO E CODIFICAÇÃO

# *Introdução*

O avanço das tecnologias de comunicação trouxe a necessidade de buscar a transmissão de informação de forma eficiente. Informação, um conceito abstrato, que, no contexto da comunicação e da tecnologia, advém da organização e interpretação de dados, fornecendo uma visão sobre tendências ou padrões, possibilitando a interpretação e emergência de significado. Dados em si, são apenas um conjunto de símbolos, geralmente organizados como uma sequência. Assim, determinada informação pode ser representada na forma de dados de maneiras distintas. A representação na forma de dados pode ser mais ou menos propícia a um determinado meio, sob o qual a informação será transmitida ou armazenada. Uma mesma informação pode ainda ser representada de forma mais concisa ou prolixa.

Se voltarmos na história, em meados do século XIX, Samuel Morse e Alfred Vail propuseram o código Morse, como uma forma econômica de se comunicar através das redes telegráficas. A proposta consistia em usar um código de pontos (tom curto), traços (tom longo) e espaços de separação entre eles (silêncio) para tornar a comunicação de uma mensagem mais eficiente, ou seja, utilizando menos pulsos e, assim, reduzindo o tempo necessário para enviar uma mensagem. O código Morse, criado originalmente para o inglês, representa os símbolos mais frequentes através de sequências curtas e os símbolos infrequentes através de sequências longas, o que proporciona a redução do comprimento esperado da mensagem transmitida.

Também no século XIX, foi criada a escrita noturna por Charles Barbier. Esta forma de escrita foi posteriormente adaptada por Louis Braille para criar um sistema mais simples e acessível para deficientes visuais. Cada célula de 6 pontos é capaz de representar letras (ou sequências de letras) na forma de combinações binárias. O código Braille, inicialmente proposto para o francês, foi mais tarde adaptado para outras línguas, bem como para matemática e música, dentre outras áreas. A adaptação da forma de escrita e leitura ao meio é fundamental e evidente na forma escrita tátil, sendo preponderante para garantir a eficácia da comunicação. É essencial que o leitor seja capaz de decodificar facilmente a informação ali representada. Para tanto, a

distinção de diferentes símbolos é facilitada pela clareza e simplicidade do sistema. A eficiência e a economicidade da representação são evidentes na capacidade de transmitir informações de forma compacta, reduzindo o espaço necessário.

Usualmente, quando falamos da representação de informação, pensamos na escrita como uma forma de expressá-la. Entretanto, devemos nos atentar ao fato de que certas informações podem utilizar outros formatos representacionais. Por exemplo, a cotação de uma moeda é representada por uma sequência de números; um sinal eletrocardiográfico é medido pela diferença de potencial; uma imagem digital é constituída por uma matriz de pixels; e informações sobre produtos, endereços ou dados de pagamento podem ser armazenados em códigos de barras e códigos QR. Embora a natureza representacional dessas informações, ao serem geradas, não seja expressa por meio de um alfabeto convencional, elas podem ser transcodificadas para serem representadas através de um alfabeto padrão, como o Base64<sup>1</sup>, que é utilizado para codificar, por exemplo, anexos de e-mails.

A ideia de representação binária é muito antiga, possivelmente anterior aos estudos de de Thomas Harriot e Gottfried Leibniz, nos séculos XVI e XVII. No entanto, Hartley (1928) foi um dos primeiros a quantificar a informação introduzindo o conceito de ‘bit’ como a unidade básica. Os trabalhos de Hartley ocorreram no contexto de rápida expansão das tecnologias de comunicação, como o telégrafo e o rádio, onde havia uma crescente necessidade de medir e otimizar a transmissão de dados.

O trabalho de Shannon (1948), que começou a ser desenvolvido durante a Segunda Guerra Mundial, permaneceu sigiloso devido à sua aplicação em comunicações militares. Embora suas ideias tenham sido formuladas na década de 1940, o artigo seminal *A Mathematical Theory of Communication* foi publicado apenas em 1948. Motivado pela necessidade de entender como a informação poderia ser codificada e transmitida de forma eficiente, Shannon desenvolveu uma teoria matemática que abordava questões práticas da comunicação. Ele introduziu conceitos fundamentais, como a entropia, que mede a incerteza ou a quantidade de informação em uma mensagem, e a capacidade do canal, que determina a quantidade máxima de informação que pode ser transmitida sem erro. O trabalho de Shannon estabeleceu um novo campo de estudo que influenciou profundamente a computação, a teoria da comunicação e outras disciplinas. Seu trabalho é considerado o marco de surgimento da teoria da informação.

Na década de 1960, a teoria da codificação passou por avanços significativos que moldaram a forma como entendemos e aplicamos a comunicação digital. O trabalho de Hamming (1950) estabeleceu as bases para a detecção e correção de erros, permitindo que sistemas de

<sup>1</sup> Base64 é um esquema de codificação que transforma dados binários em uma representação textual utilizando um conjunto de 64 caracteres, que inclui letras maiúsculas (A-Z), letras minúsculas (a-z), dígitos (0-9) e os símbolos ‘+’ e ‘/’. Cada grupo de três bytes de dados binários é convertido em quatro caracteres ASCII.

comunicação se tornassem mais robustos. Paralelamente, os códigos de Golay (1949) emergiram como uma solução eficaz para a correção de múltiplos erros, ampliando as possibilidades de transmissão confiável. A aplicação do teorema de Shannon sobre a capacidade do canal continuou a ser explorada, fornecendo uma compreensão crítica dos limites da comunicação eficiente. Além disso, os códigos de convolução começaram a ganhar destaque, oferecendo novas abordagens para melhorar a confiabilidade na transmissão de dados. O trabalho inovador de Gallager (1962), com os códigos de paridade de baixa densidade, introduziu uma nova classe de códigos que se mostraram extremamente eficazes na correção de erros, influenciando profundamente o desenvolvimento da teoria da codificação. Esses avanços não apenas solidificaram a teoria da codificação como um campo essencial da teoria da informação, mas também tiveram um impacto duradouro em diversas aplicações práticas na comunicação moderna.

A teoria da informação tornou-se central nas comunicações digitais, servindo como a base para o desenvolvimento de tecnologias que transformaram a forma como nos comunicamos. Com a ascensão da internet e das redes de comunicação, os princípios estabelecidos por Claude Shannon, como a quantificação da informação e a capacidade dos canais, tornaram-se indispensáveis para otimizar a transmissão de dados e garantir a integridade das comunicações. Além de seu papel fundamental nas telecomunicações, a teoria da informação é amplamente aplicada em diversas outras áreas. Na ecologia, por exemplo, é utilizada para analisar a diversidade de espécies e a complexidade dos ecossistemas, ajudando a entender as interações entre organismos. Na criptografia, os conceitos de entropia e codificação são essenciais para garantir a segurança das informações transmitidas. Na linguística, a teoria da informação auxilia na análise da estrutura e da semântica das línguas, permitindo a modelagem de padrões de comunicação. Outros campos, como a biologia, onde a informação genética é estudada, e a psicologia, que investiga a percepção e a cognição, também se beneficiam dos princípios da teoria da informação, demonstrando sua relevância e aplicabilidade em diferentes áreas.

# *Aleatoriedade*

A aleatoriedade é um conceito fundamental em diversas disciplinas, sendo essencial entender o que ela realmente é. Aleatoriedade é vital em diversas aplicações, sobretudo em áreas como criptografia, simulações computacionais e teoria da informação. No entanto, quando lidamos com computadores, a produção de números verdadeiramente aleatórios é um desafio intrínseco. Em geral, os números gerados por algoritmos computacionais são *pseudo-aleatórios*, ou seja, seguem sequências determinísticas que apenas aparentam não possuir uma ordem subjacente. Esses números são o resultado de funções matemáticas bem definidas, como geradores lineares congruentes ou algoritmos mais sofisticados, como o Mersenne Twister, que, apesar de produzirem sequências com propriedades estatísticas desejáveis, carecem da imprevisibilidade essencial à aleatoriedade genuína.

Stephen Wolfram, renomado por suas contribuições à complexidade computacional e à teoria dos sistemas complexos, propõe três fontes distintas de aleatoriedade que explicam como ela pode emergir em sistemas: a aleatoriedade do ambiente, a aleatoriedade nas condições iniciais e a geração intrínseca de aleatoriedade. A *aleatoriedade do ambiente* surge de influências externas imprevisíveis, como variações naturais em sistemas climáticos, que injetam incerteza contínua em um sistema. A *aleatoriedade nas condições iniciais* reflete a sensibilidade de sistemas caóticos a pequenas perturbações iniciais, exemplificada pelo “efeito borboleta”, onde mudanças mínimas podem levar a resultados drasticamente distintos. Já a *geração intrínseca de aleatoriedade* ocorre nas dinâmicas internas de sistemas determinísticos complexos, que, por meio de interações não-lineares, produzem comportamentos aparentemente aleatórios, como o ruído quântico em circuitos eletrônicos. Wolfram argumenta que essas fontes frequentemente coexistem, e, em suas palavras, até “programas simples podem produzir comportamento aparentemente aleatório mesmo sem entrada aleatória alguma”, sugerindo que os mecanismos subjacentes à aleatoriedade na natureza e em simulações computacionais compartilham uma essência comum (Wolfram 2002).

Fontes de aleatoriedade genuína, por outro lado, frequentemente

recorrem ao mundo físico, indo além dos métodos algorítmicos determinísticos. Exemplos clássicos incluem o ruído térmico em circuitos eletrônicos, a decadência radioativa ou fenômenos atmosféricos captados por sensores, usados em geradores de números aleatórios baseados em hardware. Em sistemas computacionais modernos, busca-se capturar essa aleatoriedade verdadeira por meio de eventos imprevisíveis do ambiente operacional, como os pressionamentos de teclas no teclado, os movimentos do mouse, as variações no tráfego de rede, a velocidade das ventoinhas do processador, o ruído de acesso a discos rígidos ou até mesmo as flutuações nos tempos de execução de processos do sistema. Esses dados, coletados em tempo real, são usados como sementes ou entradas diretas para aumentar a imprevisibilidade dos números gerados. Contudo, essas abordagens enfrentam limitações práticas, pois dependem de sistemas computacionais determinísticos que podem comprometer a coleta e o uso desses dados brutos ao processá-los de maneira previsível.

Um exemplo prático disso é encontrado no sistema operacional Linux, que utiliza duas fontes principais de números aleatórios: `/dev/random` e `/dev/urandom`. O dispositivo `/dev/random` gera números aleatórios de alta qualidade a partir de uma “piscina de entropia” alimentada por ruído ambiente, como os eventos mencionados, mas pode bloquear a leitura se a entropia disponível for insuficiente, tornando-o ideal para operações criptográficas que priorizam segurança sobre velocidade. Já o `/dev/urandom`, embora também utilize a mesma piscina de entropia, não bloqueia a leitura, gerando dados adicionais mesmo em baixa entropia ao reutilizar e estender os valores coletados; isso o torna mais adequado para aplicações como simulações, que demandam grandes quantidades de números aleatórios rapidamente.

### *O Gerador Lehmer*

Um dos primeiros métodos propostos para gerar números aleatórios é o método de Lehmer, proposto por Derrick Henry Lehmer na década de 1940. O método se baseia em uma relação de recorrência linear para produzir sequências de valores aparentemente aleatórios. Esse gerador, frequentemente classificado como um Gerador Linear Congruencial, utiliza a multiplicação e o módulo como peças fundamentais, definindo cada novo número na sequência por meio de:

$$X_{k+1} = g \cdot X_k \mod n, \quad (1)$$

onde  $X_k$  é o estado atual,  $g$  é um multiplicador (idealmente uma raiz primitiva módulo  $n$ ),  $n$  é o módulo (geralmente um número primo ou potência de um primo), e  $X_0$  é a semente inicial, que deve ser coprima com  $n$ .

Apesar de sua simplicidade, o Gerador Lehmer revela questões importantes sobre a percepção de aleatoriedade, tema que explorado por Stephen Wolfram em seus estudos sobre sistemas complexos. Wolfram demonstrou que a aparente aleatoriedade de uma sequência pode variar dramaticamente dependendo de como ela é visualizada. Por exemplo, ao representar os números gerados por um gerador linear congruencial como uma sequência unidimensional de valores, a saída pode parecer caótica e desprovida de padrões. No entanto, ao organizá-los em uma matriz bidimensional — onde cada valor é plotado como um ponto em linhas e colunas — ou mesmo em um cubo tridimensional, estruturas subjacentes, como repetições ou diagonais, podem emergir, revelando a natureza determinística do processo. A Figura 1 ilustra um exemplo utilizando  $g = 3$  e  $n = 2^{32}$ . Esse fenômeno

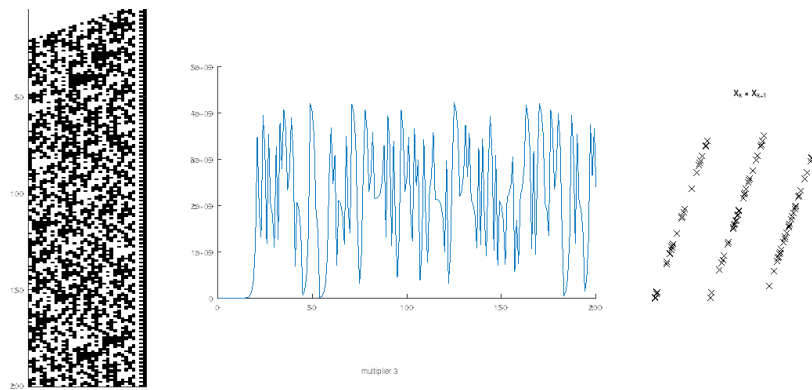


Figura 1: Representação da sequência de 200 números gerados pelo gerador Lehmer. À esquerda, a representação binária dos números, no meio um gráfico da sequência e, por fim, a representação de  $X_k \times X_{k+1}$ .

também ocorre na Regra 30 proposta por Wolfram, onde uma simples regra iterativa gera saídas que parecem aleatórias em uma dimensão, mas exibem ordem em representações visuais alternativas.

A relação entre aleatoriedade e visualização no contexto do Gerador Lehmer destaca suas limitações práticas. Se os parâmetros  $g$  e  $n$  forem mal escolhidos, a sequência pode apresentar ciclos curtos ou padrões visíveis, comprometendo sua utilidade em aplicações que exigem alta imprevisibilidade. Por exemplo, em testes gráficos como o plano de fases (plotando  $X_k$  contra  $X_{k+1}$ ), geradores lineares simples frequentemente revelam retas ou grades, evidenciando correlações indesejadas. Mesmo com parâmetros otimizados, a natureza determinística do método implica que ele nunca produz aleatoriedade genuína, apenas uma aproximação que pode ser suficiente para cenários, como simulações básicas, mas inadequada para criptografia.

### *O Gerador Mersenne Twister*

O Mersenne Twister é um algoritmo de geração de números pseudo-aleatórios desenvolvido por Makoto Matsumoto e Takuji Nishimura em 1997, amplamente reconhecido por sua capacidade de produzir sequências com propriedades estatísticas de alta qualidade e um período excepcionalmente longo. O nome “Mersenne Twister” reflete sua fundamentação matemática: o algoritmo utiliza números primos de Mersenne — números primos da forma  $2^p - 1$ , onde  $p$  é também um primo — e emprega uma técnica chamada *twisting* para garantir a geração de números com boa distribuição estatística e independência aparente. O algoritmo é projetado para prover uma distribuição uniforme e baixa correlação entre valores consecutivos.

O Mersenne Twister também oferece um equilíbrio entre qualidade estatística e eficiência computacional. Embora não seja o gerador mais rápido disponível — métodos mais simples, como geradores lineares congruenciais, podem ser mais rápidos em termos de tempo de execução —, ele compensa com sua robustez e confiabilidade, tornando-o adequado para uma ampla gama de aplicações. Por essa

O Mersenne Twister também possui uma boa eficiência computacional, sendo por isso amplamente adotado em linguagens de programação, como a função `rand` do Octave e do MATLAB, que utiliza a variante MT19937 com período  $2^{19937} - 1$  (seu período extremamente longo), bem como em bibliotecas padrão de linguagens como Python (`random`) e C++ (`std::mt19937`). Apesar de suas qualidades, o Mersenne Twister é inadequado para aplicações criptográficas.

### *MT19937*

O Mersenne Twister (MT) (Matsumoto e Nishimura 1998) é um dos geradores de números pseudoaleatórios (RNGs) mais amplamente utilizados, sendo um GFSR (*Generalized Feedback Shift Register*) com *twist*. Conforme Tan (2016), o algoritmo do MT pode ser descrito pelas seguintes etapas:

- **Inicialização:** Defina  $r$  números de  $w$  bits ( $x_i, i = 1, 2, \dots, r$ ) aleatoriamente como valores iniciais.
- **Definição da Matriz de Recorrência:** Considere a matriz

$$A = \begin{pmatrix} 0 & I_{w-1} \\ a_w & a_{w-1} \cdots a_1 \end{pmatrix},$$

onde  $I_{w-1}$  é a matriz identidade  $(w-1) \times (w-1)$ , e  $a_i, i = 1, \dots, w$ , assume valores 0 ou 1.



- **Recorrência:** Para cada  $i \geq 0$ , calcule

$$x_{i+r} = (x_{i+s} \oplus (x_i^{(w:(l+1))} | x_{i+1}^{(l:1)}) A),$$

onde  $x_i^{(w:(l+1))} | x_{i+1}^{(l:1)}$  representa a concatenação dos  $w - l$  bits mais significativos (superiores) de  $x_i$ , e  $x_{i+1}^{(l:1)}$  os  $l$  bits menos significativos de  $x_{i+1}$ .

- **Tempering:** Para cada  $x_{i+r}$ , aplique as operações sequencialmente:

$$\begin{aligned} z &= x_{i+r} \oplus (x_{i+r} \ggg t_1), \\ z &= z \oplus ((z \lll t_2) \wedge m_1), \\ z &= z \oplus ((z \lll t_3) \wedge m_2), \\ z &= z \oplus (z \ggg t_4), \\ u_{i+r} &= z / (2^w - 1), \end{aligned}$$

onde  $t_1, t_2, t_3, t_4$  são inteiros,  $m_1, m_2$  são máscaras de bits, e  $\wedge$  é a operação *bitwise AND*. Os valores  $u_{i+r}, i = 1, 2, \dots$ , formam a sequência no intervalo  $[0, 1]$ .

Com valores apropriados de  $r$ , o MT pode gerar sequências com período de até  $2^{19937}$ , além de possuir boas propriedades estatísticas (Matsumoto e Nishimura 1998).

### *Abordagens para Geração de Números Pseudoaleatórios*

Diversas abordagens existem para gerar números pseudoaleatórios (Knuth 1997; Matsumoto e Nishimura 1998; Wolfram 2002; Kocarev e Lian 2011; Kneusel 2018), abaixo listamos algumas delas:

- Geradores Lineares Congruenciais (LCGs):  
Baseiam-se em uma relação de recorrência linear da forma

$$X_{n+1} = (a \cdot X_n + c) \mod m, \quad (2)$$

como o exemplo do Gerador Lehmer visto anteriormente. Este tipo de gerador é simples, mas com período curto e baixa qualidade estatística.

- Geradores de Fibonacci com Atraso (LFGs):  
Geram números combinando valores anteriores da sequência usando uma operação (como adição ou XOR) com atrasos:

$$X_n = (X_{n-k} \oplus X_{n-l}) \mod 2^w, \quad (3)$$

oferecendo períodos longos, mas exigindo estados iniciais grandes.

- Autômatos Celulares:

Aplicam regras como a Regra 30 vista anteriormente,

$$c_i(t+1) = c_{i-1}(t) \oplus (c_i(t) \vee c_{i+1}(t)), \quad (4)$$

gerando padrões caóticos, porém computacionalmente custosos.

- Funções de Hash e Criptográficas:

Utilizam funções como SHA-256 ( $R_i = H(S||C)$ ) ou cifras de fluxo como ChaCha ( $R_i = \text{ChaCha}(K, C)$ ), sendo seguras para criptografia, porém mais lentas.

- Geradores Baseados em Caos:

Empregam sistemas caóticos, como o mapa logístico

$$X_{n+1} = r \cdot X_n \cdot (1 - X_n), \quad (5)$$

com comportamento caótico, mas sensíveis a precisão numérica.

## *Princípios Fundamentais*

A comunicação envolve alguns componentes fundamentais: uma fonte, que gera a mensagem; um transmissor, que envia a mensagem através de algum meio; um canal de comunicação, que é o meio pelo qual a comunicação se estabelece, sendo suscetível a ruídos e interferências; um receptor, que recebe a mensagem; e, por fim, o destinatário da mensagem (veja o diagrama na Figura 2, adaptado de “*A Mathematical Theory of Communication*” Shannon 1948). No âmago do processo de comunicação reside um problema fundamental: reconstruir no receptor a exata mensagem pretendida pelo emissor. Independentemente de estarmos lidando com a comunicação falada, escrita ou por meio de sinais digitais, o objetivo é o mesmo. Embora cada sistema guarde suas nuances, a teoria da informação proposta por Shannon (1948) busca lidar com essa problemática sob uma perspectiva unificada. Para tanto, ele introduziu conceitos fundamentais como entropia e capacidade de canal. Neste livro, procuraremos apresentar uma visão geral da teoria, seus fundamentos e aspectos práticos.

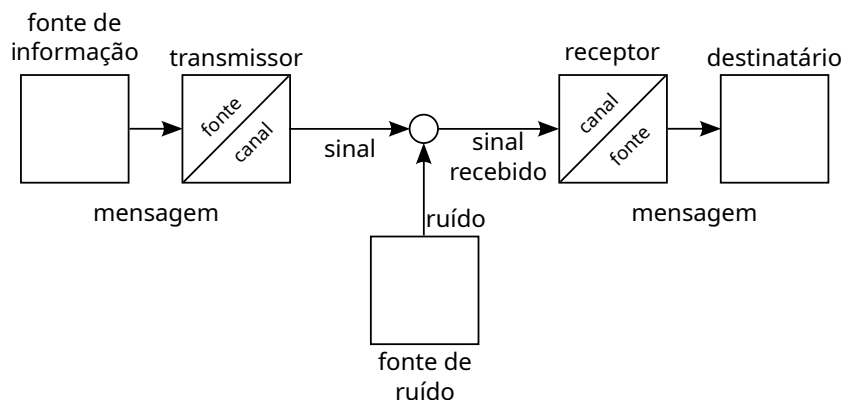


Figura 2: Diagrama genérico de um sistema de comunicações.

O diagrama ilustrado na Figura 2 apresenta a sistematização proposta por Shannon (1948), incluindo conceitos e aspectos da comunicação já utilizados à época, mas trazendo uma abordagem sistemática para o conceito de comunicação. Além disso, ele divide a tarefa do codificador e decodificador em duas partes distintas: a codifica-

ção/decodificação de fonte e a codificação/decodificação de canal. Essa separação permite abordar cada uma dessas partes do sistema de comunicação de forma independente, possibilitando, assim, projetar e otimizar cada uma delas separadamente.

Para lidar matematicamente com a comunicação, um processo que envolve incerteza e variabilidade, e trazer então uma perspectiva probabilística na abordagem do problema, inicialmente, é essencial entender o conceito de variável aleatória (v.a.). Uma variável aleatória é uma variável que pode assumir diferentes valores dependendo de fatores aleatórios, ou seja, opera-se um mecanismo não determinístico que torna impossível prever seu valor. Ela representa a incerteza inerente a eventos ou processos aleatórios e pode ser utilizada para modelar diversos experimentos, como o lançamento de um dado, a previsão do tempo ou a sequência de caracteres em um texto. Na notação aqui adotada, as variáveis aleatórias serão representadas por letras maiúsculas, como, por exemplo,  $X$ . Uma variável aleatória discreta é aquela que pode assumir valores em um conjunto enumerável. As realizações de uma v.a. se dão de acordo com uma distribuição subjacente  $p$ , e assim dizemos que  $X \sim p$ . Para representar o conjunto de valores que a variável aleatória pode assumir, utilizaremos a letra em forma caligráfica  $\mathcal{X}$ , enquanto um valor específico que a variável pode assumir será denotado por uma letra minúscula, como  $x$ . Assim, podemos descrever que a variável aleatória  $X$  assume o valor  $x$  através do evento  $\{X = x\}$ .

Dada uma v.a.  $X$ , em um alfabeto  $\mathcal{X}$  de cardinalidade  $N = |\mathcal{X}|$ , onde  $\mathcal{X} = \{a_1, \dots, a_N\}$  e  $a_i$ ,  $i = 1, \dots, N$ , representa cada um dos possíveis valores que a v.a. pode assumir com probabilidade  $p_i$ , ou seja,  $p_i = \Pr(X = a_i)$ . A distribuição subjacente que rege a v.a.  $X$  é  $p = \mathcal{P}_X = \{p_1, \dots, p_N\}$ , tal que  $p_i \geq 0$  e  $\sum_{i=1}^N p_i = 1$  ou, de forma equivalente,  $\sum_{x \in \mathcal{X}} \Pr(X = x) = 1$ .

Hartley (1928) introduziu a ideia de que a quantidade de informação pode ser medida em termos de possibilidades. Hartley propôs que a informação é diretamente proporcional ao logaritmo do número de estados possíveis de um sistema, estabelecendo assim uma base para a quantificação da informação. A medida de informação associada a uma variável aleatória  $X$ , com alfabeto de tamanho  $N$ , é expressa por

$$I(X) = \log_b L, \quad (6)$$

onde  $b$  é a base utilizada para medir tal informação. Para  $b = 2$ , a informação será medida em ‘bits’ (nome sugerido por J.W. Tukey).

O conceito proposto por Hartley sobre a quantificação da informação é diretamente consonante com a interpretação da capacidade representacional de uma unidade de memória. Em termos práticos, uma unidade de memória com  $n$  bits é capaz de representar  $2^n$  se-

quências binárias distintas, o que implica que essa unidade possui  $n$  bits de informação. Quando consideramos duas unidades de memória, cada uma com  $n$  bits, a capacidade total se torna  $2n$  bits, permitindo a representação de  $2^{2n}$  sequências binárias distintas. Essa relação demonstra que, conforme a quantidade de bits de memória aumenta, a capacidade de representação de informações também cresce exponencialmente, alinhando-se perfeitamente com a proposta de Hartley de que a informação é medida em função do número de estados possíveis.

Um conceito fundamental que emerge na discussão sobre a teoria da informação é a redundância, ou ainda, a previsibilidade. A redundância refere-se à parte da informação que é repetitiva ou desnecessária para a compreensão, podendo ser eliminada sem perda de significado. Enquanto, sob uma primeira vista, pode ser vista como um desperdício de recursos, a redundância também desempenha um papel crucial na detecção e correção de erros, permitindo que a informação seja transmitida de forma mais confiável em ambientes ruidosos. Exemplo disso é a redundância presente na comunicação escrita. Veja como o trecho a seguir pode ser facilmente compreendido, mesmo com lacunas: “Nda se mudria, o regim, si era posívl, ms tmbém s muda d roupa sm trear d pele.”<sup>2</sup> A capacidade de compreender o trecho, apesar das palavras truncadas e da ausência de letras, demonstra que a linguagem é intrinsecamente projetada para lidar com a incerteza e a imperfeição no processo de transmissão. Se não houvesse redundância, qualquer perda de informação tornaria a decodificação da mensagem inviável, resultando em confusão e mal-entendidos.

Shannon, por outro lado, incorporou o conceito de redundância em sua análise da informação. Ao considerar eventos com probabilidades distintas, é importante notar que a incerteza associada a esses eventos não pode ser a mesma. Assim, a incerteza não depende apenas do tamanho do alfabeto, mas também das probabilidades associadas a cada evento. Para um evento  $E_k$  com probabilidade  $p_k$ , a quantidade de informação (medida em bits) associada a esse evento é expressa pela fórmula

$$I(E_k) = -\log_2(p_k). \quad (7)$$

Essa relação indica que eventos menos prováveis, ou seja, aqueles com probabilidades mais baixas, geram uma maior quantidade de informação, pois sua ocorrência é mais inesperada. Em contrapartida, eventos com alta probabilidade resultam em uma quantidade menor de informação, refletindo a redundância presente na comunicação. Essa compreensão da relação entre probabilidade e informação é essencial para o desenvolvimento do conceito de entropia, que serve como uma medida da incerteza, aleatoriedade e quantidade de informação associada a uma variável aleatória.

<sup>2</sup> Trecho original: “Nada se mudaria; o regime, sim, era possível, mas também se muda de roupa sem trocar de pele. Comércio é preciso. Os bancos são indispensáveis. No sábado, ou quando muito na segunda-feira, tudo voltaria ao que era na véspera, menos a constituição.” (cap. LXIV de Esaú e Jacó, Machado de Assis).

## Entropia

Shannon (1948) propôs a entropia com forma de quantificar a incerteza associada a uma v.a., fornecendo assim uma medida da aleatoriedade. A entropia nos permite entender como a distribuição de probabilidades de uma variável aleatória influencia na informação associada a ela. A entropia de uma v.a. é então expressa como o valor esperado da informação própria associada aos eventos no alfabeto desta v.a., ou seja,  $E_p[I(E_k)]$  e será expressa por  $H(X)$ .

**Definição 1** (Entropia).

$$H(X) \triangleq E_p[-\log(p_k)] \quad (8a)$$

$$= - \sum_{k=1}^{|\mathcal{X}|} p_k \log p_k \quad (8b)$$

$$= - \sum_{x \in \mathcal{X}} p(x) \log p(x), \quad (8c)$$

Passamos a adotar a convenção  $\log = \log_2$ , uma vez que estaremos sempre medindo a informação em bits. Quando necessário utilizar uma base diferente, expressaremos explicitamente.

Para encontrar a formulação dada na Equação (8), Shannon (1948) propôs uma função de entropia  $H$ , que depende da distribuição de massa  $p$ . Dado um alfabeto de tamanho  $N = \mathcal{X}$ , e distribuição  $p = \{p_1, \dots, p_N\}$ , Shannon estabeleceu três propriedades fundamentais que a função de entropia deveria satisfazer. A primeira propriedade exige que  $H$  seja contínua em relação às probabilidades  $p_i$ , o que significa que pequenas mudanças nas probabilidades não devem causar grandes saltos na medida de informação. A segunda propriedade afirma que, se todos os  $p_i$  são iguais, a entropia deve aumentar monotonicamente com o tamanho do alfabeto  $N$ . Isso reflete a intuição de que mais opções disponíveis para escolha devem resultar em maior incerteza e, portanto, maior entropia. Por fim, a terceira propriedade, conhecida como a propriedade da aditividade, estabelece que se uma escolha pode ser decomposta em uma sequência de escolhas independentes, a entropia total deve ser a soma ponderada das entropias individuais de cada escolha. Essa propriedade é crucial para a análise de sistemas complexos, onde a informação pode ser tratada em partes menores e combinadas para obter uma visão global.

A formulação matemática da entropia de Shannon, dada na Equação (8), emerge naturalmente ao considerar essas propriedades, sendo a função logarítmica utilizada devido à sua capacidade de transformar multiplicações em somas, o que é essencial para garantir a aditividade da medida de informação.

## Interlúdio

Este interlúdio fornece uma pausa para apresentar a demonstração contida no Anexo 2 do artigo de Shannon (1948). O leitor, se desejar, pode ir direto à página 16, onde finaliza-se este entreato.

Define-se a  $A(N)$  como o caso específico de  $H$  para uma distribuição uniforme com alfabeto de tamanho  $N$ :

$$A(N) = H\left(\frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}\right). \quad (9)$$

Deseja-se que uma escolha dentre  $s^M$  opções igualmente prováveis possa ser decomposta como uma sequência de  $M$  escolhas que se subdividem em  $s$  possibilidades igualmente prováveis.

Teremos então que

$$A(s^M) = MA(s). \quad (10)$$

Da mesma forma, para  $t$  e  $N$ , teremos  $A(t^N) = NA(t)$ . Podemos tomar  $N$  arbitrariamente grande e encontrar  $M$  que satisfaça

$$s^M \leq t^N \leq s^{(M+1)}. \quad (11)$$

Tomando o logaritmo<sup>3</sup> da expressão acima e dividindo por  $N \log s$  todos os termos<sup>4</sup>, teremos

$$\frac{M}{N} \leq \frac{\log t}{\log s} \leq \frac{M}{N} + \frac{1}{N}, \quad (12)$$

o que é equivalente a

$$\left| \frac{M}{N} - \frac{\log t}{\log s} \right| < \epsilon, \quad (13)$$

onde  $\epsilon$  é arbitrariamente pequeno, já que  $N$  é arbitrariamente grande.

Usando agora a propriedade desejada de monotonicidade de  $A(N)$ , teremos

$$A(s^M) \leq A(t^N) \leq A(s^{(M+1)}) \quad (14a)$$

$$MA(s) \leq NA(t) \leq (M+1)A(s) \quad (14b)$$

Dividindo a expressão acima por  $NA(s)$ , teremos

$$\frac{M}{N} \leq \frac{A(t)}{A(s)} \leq \frac{M}{N} + \frac{1}{N}, \quad (15)$$

ou, de forma equivalente,

$$\left| \frac{M}{N} - \frac{A(t)}{A(s)} \right| < \epsilon, \quad (16)$$

e assim, como as duas frações ( $\log t / \log s$  e  $A(t)/A(s)$ ) estão  $\epsilon$  próximas de  $M/N$ , podemos concluir que

$$\left| \frac{A(t)}{A(s)} - \frac{\log t}{\log s} \right| < 2\epsilon. \quad (17)$$

Como  $\epsilon$  é arbitrariamente pequeno, no limite teremos

$$\frac{A(t)}{A(s)} = \frac{\log t}{\log s} \quad (18a)$$

$$A(t) = \frac{A(s)}{\log s} \log t = K \log t, \quad (18b)$$

onde  $K$  deve ser positivo, de forma que  $A(N)$  seja monótona crescente.

Suponha uma escolha com  $N$  possibilidades em que as probabilidades são comensuráveis,  $p_i = N_i / \sum N_i$ , onde  $N_i$  são inteiros. De forma equivalente,

<sup>3</sup> Logaritmo é uma função monótona crescente.

<sup>4</sup>  $N \log s$  é positivo para  $N \geq 0$  e  $s \geq 1$ .

uma escolha entre  $\sum N_i$  opções pode ser expressa como uma escolha dentre  $N$  opções com probabilidades  $p_1, \dots, p_N$ , e para uma  $i$ -ésima dada escolha, realizar uma nova escolha dentre  $N_i$  opções igualmente prováveis. Teremos então:

$$\overbrace{K \log \left( \sum N_i \right)}^{A(\sum N_i)} = H(p_1, \dots, p_N) + \overbrace{K \log N_i}^{A(N_i)} \quad (19a)$$

$$K \underbrace{\left( \sum p_i \right)}_{=1} \log \left( \sum N_i \right) = H(p_1, \dots, p_N) + K \underbrace{\left( \sum p_i \right)}_{=1} \log N_i. \quad (19b)$$

E assim,

$$H(p_1, \dots, p_N) = K \left[ \left( \sum p_i \right) \log \left( \sum N_i \right) - \left( \sum p_i \right) \log N_i \right] \quad (20a)$$

$$= -K \sum p_i \log \frac{N_i}{\sum N_i} = -K \sum p_i \log p_i. \quad (20b)$$

□

Tomemos primeiramente um exemplo simples de uma v.a. binária  $X \in \mathcal{X} = \{0, 1\}$ . Para simplificar, iremos denotar arbitrariamente por  $\theta = Pr(X = 1)$  e  $1 - \theta = Pr(X = 0)$ . Usando agora a definição dada na Equação (8), teremos

$$H(X) = -\theta \log \theta - (1 - \theta) \log(1 - \theta) \quad (21)$$

e, para simplificar, iremos denotar esta grandeza por  $H(\theta)$ . Sempre que utilizarmos esta notação, como por exemplo em  $H(1/4)$ ,  $H(1/3)$  ou  $H(\pi)$ , estaremos nos referindo à entropia binária definida na Equação (21).

A Figura 3 apresenta o gráfico da entropia binária dada pela Equação (21). Note como  $H = 0$  quando  $\theta = 1$  ou  $\theta = 0$ , ou seja, nos casos em que um dos eventos ocorre com probabilidade 1. Observe ainda como o máximo é alcançado em  $\theta = 1/2$ , quando ambos eventos são equiprováveis (máxima incerteza). A função observada é côncava, o que, mais à frente, mostraremos ser válido para toda função de entropia. A concavidade garante ainda que ao misturarmos (ponderarmos) distribuições diferentes não é possível obter uma entropia menor que a ponderação das entropias das partes.

Vamos considerar agora o exemplo de uma v.a. com distribuição uniforme  $X \sim u$ . Usando a definição na Equação (8), teremos que a entropia de  $X$  será dada por

$$H(X) = - \sum_{k=1}^{|\mathcal{X}|} \frac{1}{|\mathcal{X}|} \log \frac{1}{|\mathcal{X}|} \quad (22a)$$

$$= -\log \frac{1}{|\mathcal{X}|} = \log |\mathcal{X}|. \quad (22b)$$

Como esperávamos, a entropia de uma distribuição uniforme é monotonicamente crescente com o tamanho do alfabeto. Intuitivamente,

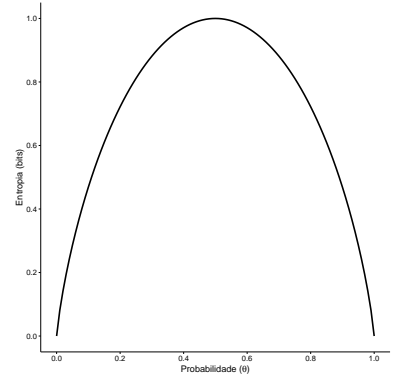


Figura 3: Entropia binária



como já mencionado, a entropia é máxima para uma distribuição uniforme, uma vez que é a situação de maior incerteza para um dado alfabeto.

Ao analisarmos a Equação (8), podemos nos questionar o que ocorre quando algum símbolo tem probabilidade zero, uma vez que aparecerá um termo  $0 \log 0$  no somatório. Neste caso, iremos usar o valor limite  $\lim_{x \rightarrow 0} x \log x = 0$ . Para demonstrar (a menos de uma constante pela mudança de base do logaritmo), usaremos para tanto a regra de L'Hospital:

$$\lim_{x \rightarrow 0} x \ln x = \lim_{x \rightarrow 0} \frac{\ln x}{1/x} \quad (23a)$$

$$= \lim_{x \rightarrow 0} \frac{1/x}{-1/x^2} \quad (23b)$$

$$= \lim_{x \rightarrow 0} -x = 0. \quad (23c)$$

No próximo exemplo, vamos buscar aproximar a entropia de textos escritos. Sabemos que a distribuição das palavras em textos segue uma lei de potência conhecida como Lei de Zipf Zipf 1935; Zipf 1949; Ferrer i Cancho e Solé 2001; Araújo, Cristófar-Silva e Yehia 2013. A Lei de Zipf é uma observação empírica que também é observada em diversos outros naturais. Ao analisar um corpus de texto, Zipf constatou que a frequência de uma palavra é inversamente proporcional à sua posição em um ranking de frequência. Essa relação pode ser expressa matematicamente como

$$p_k(s, N) = C k^{-s}, \quad (24)$$

onde  $p_k$  é a probabilidade de ocorrência da  $k$ -ésima palavra mais frequente,  $C$  é uma constante de normalização ( $C^{-1} = \sum_{n=1}^N n^{-s}$ ),  $k$  é o *rank*,  $s$  a constante que caracteriza a distribuição e  $N$  o tamanho do léxico. A Figura 4 apresenta o gráfico de Zipf (frequência/probabilidade vs. *rank*) para a obra completa de William Shakespeare, dividida em 44 textos, conforme organizados no Projeto Gutenberg<sup>5</sup>. Cada uma das série de pontos na Figura 4 representa um destes 44 textos. Note como o comportamento observado entre eles é bem similar.

<sup>5</sup> <https://www.gutenberg.org/>

Usando a Equação (24) em conjunto com a Equação (8), obtemos a formulação da entropia para uma distribuição de Zipf:

$$H(X) = s C \sum_{k=1}^N \frac{\log k}{k^s} - \log C. \quad (25)$$

Após estimar a entropia de cada um dos textos de Shakespeare, podemos observar no histograma da Figura 5 como a concentração dos valores encontrados está por acima de 9 bits. Como observaremos mais

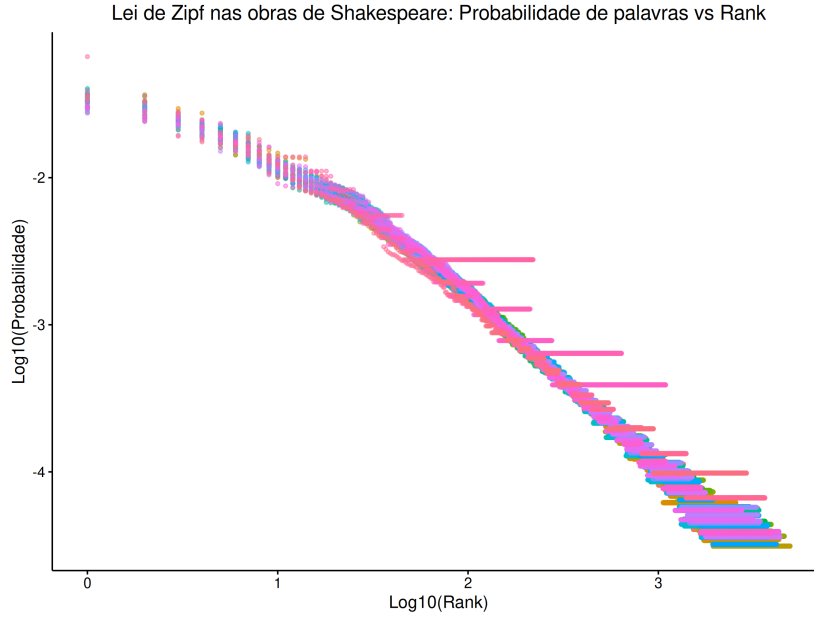


Figura 4: Gráfico de Zipf para 44 textos de William Shakespeare no Projeto Gutenberg.

à frente, conhecer o valor da entropia é importante para sabermos qual é o limite representacional para uma fonte (qual é o limite da compressão). Além disso, linguistas utilizam a entropia para analisar a complexidade de uma língua, redundância e eficiência do sistema de escrita.

### *Propriedades da Entropia*

A entropia possui várias propriedades fundamentais que iremos descrever a seguir. Essas propriedades não apenas ajudam a entender o comportamento da entropia e medidas relacionadas, como também serão úteis em diversas demonstrações. Nesta seção, exploraremos as principais propriedades da entropia, incluindo a não negatividade, a concavidade, valor máximo, a continuidade, e a mudança de base.

Como dito anteriormente, usualmente a unidade escolhida para medir a informação é bit, e portanto utilizamos a base 2. Entretanto, a entropia pode ser dada também em outras bases: nats (base  $e$ ), trits (base 3), hartley (base 10). Para obtermos a entropia em outras bases, basta utilizar a regra de mudança de base do logaritmo:  $\log_a x = \log_b x / \log_b a$ . Teremos assim  $H_b(X) = (\log_b a) H_a(X)$ . A transformação para a base  $e$  costuma ser útil nas demonstrações para evitar o fardo de se carregar uma constante durante todo o seu percurso.

Uma propriedade muito importante é a não negatividade, ou seja,  $H(X) \geq 0$ . A incerteza nunca<sup>6</sup> será negativa, o que faz sentido intuitivo.

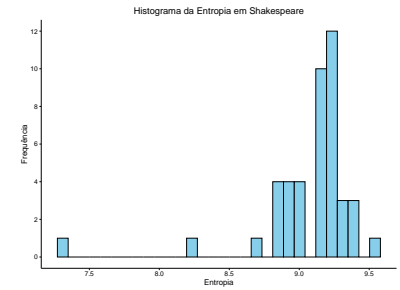


Figura 5: Histograma da entropia nos 44 textos de William Shakespeare.

<sup>6</sup> Importante ressaltar que estamos aqui tratando de variáveis discretas. No contexto contínuo a entropia pode ser negativa e terá outra interpretação.

tivamente. Além disso, a não negatividade é uma propriedade importante para encontrarmos outras relações. Para verificá-la, basta notar que a entropia é a soma de termos da forma  $-p \log p$ , como  $0 \leq p \leq 1$ , teremos que cada termo da soma é maior ou igual a zero. Por conseguinte, a entropia seria não negativa.

Para verificar que a entropia é uma função contínua em  $p$ , devemos primeiramente definir  $l(x)$  como a extensão contínua de  $x \log x$ , da seguinte forma:

$$l(x) = \begin{cases} x \log x & \text{se } x > 0, \\ 0 & \text{se } x = 0, \end{cases} \quad (26)$$

e assim,  $H(X) = H(p)$  é definido como

$$H(p) = - \sum_{x \in \mathcal{X}} l(p_x). \quad (27)$$

Sendo a entropia definida como um somatório finito de funções contínuas, fica evidente que ela também será uma função contínua.

Antes de abordarmos uma próxima propriedade fundamental da entropia, vamos estabelecer uma desigualdade simples e importante em diversas demonstrações futuras. Ela é apresentada graficamente na Figura 6 e demonstrada formalmente no Teorema 1.

**Lema 1** (Desigualdade Fundamental) *Para qualquer  $z > 0$ ,*

$$\ln z \leq z - 1, \quad (28)$$

*com igualdade se e somente se  $z = 1$ .*

*Demonstração.* Sabemos que para  $z = 1$  é verdadeiro, pois  $0 = \ln 1 \leq 1 - 1 = 0$ . Vamos demonstrar que  $\ln z \leq z - 1$ , para  $z \geq 1$  por contradição. Suponha que existe  $b > 1$  tal que  $\ln z > z - 1$ , para  $z = b$ . Vamos definir  $f(z) = \ln z - z + 1$ , logo  $f(1) = 0$  (conforme visto acima) e  $f(b) > 0$  (por hipótese). Pelo teorema do valor médio  $\exists c, 1 < c < b$ , tal que

$$f'(c) = \frac{f(b) - f(1)}{b - 1} = \underbrace{\frac{f(b)}{b - 1}}_{>0} > 0. \quad (29)$$

Mas,  $f'(z) = 1/z - 1$ , e assim  $f'(z) < 0$  para  $z > 1$ . Logo há uma contradição e nossa hipótese é falsa. Teremos assim  $\ln z \leq z - 1$ , para  $z \geq 1$ .

Para  $z \in (0, 1)$ , basta seguir os mesmos passos, escolhendo um ponto  $z = b$ , tal que  $0 < b < 1$ . Vamos encontrar um ponto  $c$  tal que  $0 < b < c < 1$ . E usar o teorema do valor médio para mostrar uma contradição na hipótese.

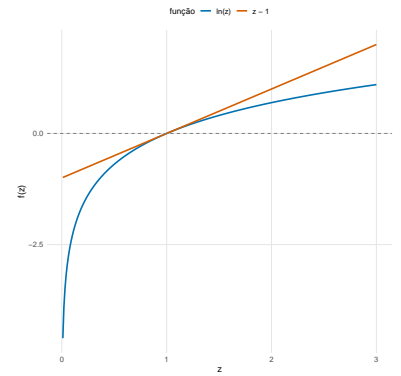


Figura 6: Demonstração gráfica da desigualdade  $\ln z \leq z - 1$  (embora não seja uma demonstração formalmente válida, serve como intuição).

□

Outra propriedade importante que mencionamos é o limite máximo da entropia, a entropia da distribuição uniforme. Note que, mostrar que  $H(X) \leq \log N$  é equivalente a mostrar que  $H(X) - \log N \leq 0$ . Para tanto, vejamos:

$$H(X) - \log N = - \sum_x p(x) \log p(x) - \log N \overbrace{\sum_x p(x)}^{=1} \quad (30a)$$

$$= - \sum_x p(x) \log p(x) - \sum_x p(x) \log N \quad (30b)$$

$$= - \sum_x p(x) \log p(x) N \quad (30c)$$

$$= \log_2 e \sum_x p(x) \ln \frac{1}{p(x)N} \quad (30d)$$

$$\leq \log_2 e \sum_x p(x) \left[ \frac{1}{p(x)N} - 1 \right] \quad (30e)$$

$$= \log_2 e \left[ \underbrace{\sum_{x \in \mathcal{X}} \frac{1}{N}}_{=N \frac{1}{N}=1} - \underbrace{\sum_x p(x)}_{=1} \right] = 0. \quad (30f) \quad \square$$

Na demonstração anterior, utilizamos a Equação (28), com  $z = 1/p(x)N$ . A igualdade  $\ln z = z - 1$  se dará no ponto  $z = 1$ , isto é, quando  $1/p(x)N = 1$ , ou seja, quando  $p(x) = 1/N$ , e teremos assim uma distribuição uniforme.

A concavidade da entropia é uma propriedade importante. Garante que a entropia terá um único máximo global. A maximização da entropia é um princípio fundamental em várias áreas como estatística, para a inferência de distribuições de probabilidade, e aprendizado de máquina, para garantir que os algoritmos de otimização converjam e para garantir a estabilidade dos modelos. Esta propriedade será demonstrada mais à frente.

### *Entropia Conjunta e Entropia Condicional*

Quando lidamos com incerteza em situações que envolvem múltiplas variáveis aleatórias, entropia conjunta e a entropia condicional são conceitos que surgem da extensão das definições vistas anteriormente. Um par de variáveis aleatórias  $(X, Y)$  pode ser visto como uma variável aleatória vetorial.

A entropia conjunta de duas variáveis aleatórias  $X$  e  $Y$ , denotada como  $H(X, Y)$ , mede a incerteza total associada ao par de variáveis. É definida como:

**Definição 2** (Entropia Conjunta).

$$H(X, Y) \triangleq - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y) \quad (31a)$$

$$= \mathbb{E}_{p(x, y)} \log \frac{1}{p(X, Y)} \quad (31b)$$

$$= \mathbb{E} \log \frac{1}{p(X, Y)}, \quad (31c)$$

onde, em 31c, para simplificar<sup>7</sup>, omitimos a distribuição em respeito a qual o valor esperado é tomado.

<sup>7</sup> Tais simplificações pode aparecer ao longo do texto. Porém as adotaremos apenas em casos que não gerem ambiguidades.

Generalizando para um vetor de variáveis aleatórias  $X_{1:N} = (X_1, X_2, \dots, X_N)$ :

$$H(X_{1:N}) = H(X_1, X_2, \dots, X_N) \quad (32a)$$

$$= \sum_{x_1, x_2, \dots, x_N} p(x_1, \dots, x_N) \log \frac{1}{p(x_1, \dots, x_N)} \quad (32b)$$

$$= \mathbb{E} \log \frac{1}{p(X_1, \dots, X_N)}. \quad (32c)$$

Dada a definição, vamos analisar um exemplo prático para analisar a interdependência entre caracteres consecutivos em uma língua natural. Para calcular a entropia conjunta de dois caracteres em sequência, devemos primeiramente calcular a probabilidade conjunta. A Figura 7 apresenta o resultado encontrado para a obra completa de Shakespeare. Note como os maiores valores correspondem às sequências ‘th’, ‘he’, ‘er’, ‘an’, dentre outras (devido às palavras de alta frequência de ocorrência na língua, como “the”, “in”, “her”, “there”, etc.

Ao calcular a entropia conjunta encontramos  $H(X, Y) = 7.8482$  bits, o que é menor do que o dobro da entropia de um único caractere,  $2 \times H(X) = 8.3798$  bits. O fato da entropia conjunta de dois caracteres consecutivos ser menor do que o dobro da entropia de um único caractere indica que os caracteres não são independentes: a ocorrência de um caractere influencia a probabilidade do próximo, reduzindo a incerteza total. Em uma língua natural, essa interdependência é esperada devido a padrões linguísticos.

Essa interdependência sugere que a incerteza associada ao segundo caractere ( $Y$ ), dado o conhecimento do primeiro caractere ( $X$ ), é menor do que a incerteza do segundo ( $Y$ ) quando considerado isoladamente. Em outras palavras, o conhecimento de  $X$  fornece informação que reduz a incerteza sobre  $Y$ , como ocorre em pares comuns como ‘th’, onde ‘t’ aumenta a probabilidade de ‘h’. Essa incerteza reduzida é formalmente chamada de entropia condicional, denotada por  $H(Y|X)$ , e é definida como

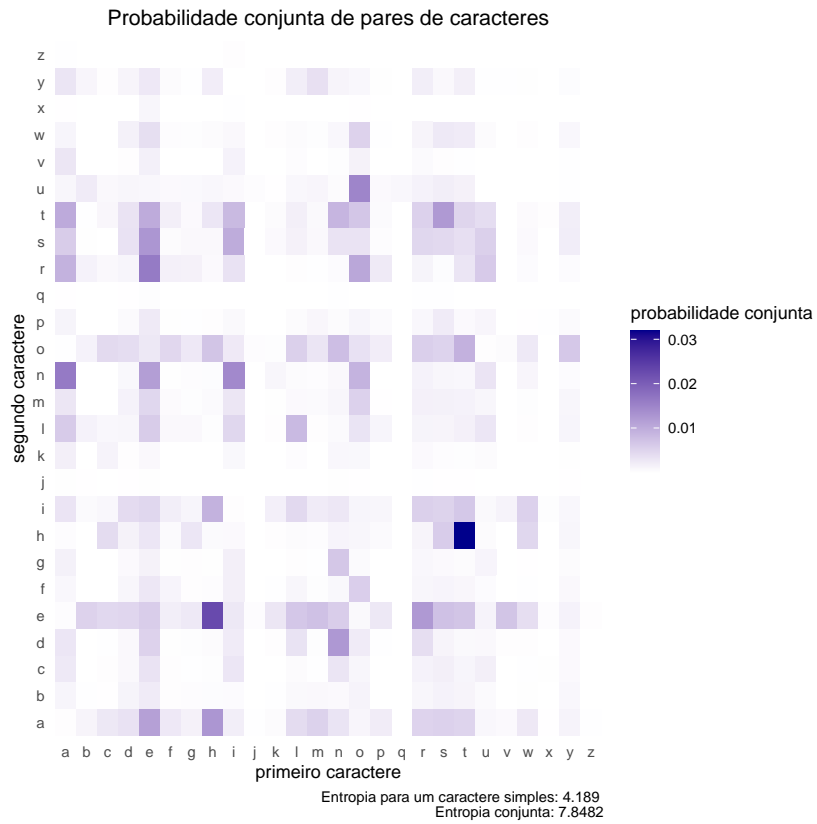


Figura 7: Probabilidade conjunta do primeiro e segundo caractere analisando os 44 textos de William Shakespeare no Projeto Gutenberg. As entropias simples e conjunta são apresentadas, evidenciando que a entropia conjunta é maior, porém, analisando a entropia por caractere, a entropia conjunta abarca dependências entre caracteres consecutivos, reduzindo o efeito de incerteza por caractere.

**Definição 3** (Entropia Condicional).

$$H(Y|X) \triangleq \sum_x p(x) H(Y|X=x) \quad (33a)$$

$$= - \sum_x p(x) \sum_y p(y|x) \log p(y|x) \quad (33b)$$

$$= - \sum_{x,y} p(x,y) \log p(y|x) \quad (33c)$$

$$= E_{p(x,y)} \log \frac{1}{p(Y|X)}. \quad (33d)$$

A definição dada pela Equação (33a) nos mostra como a entropia condicional de  $Y$ , dado  $X$ , é o valor esperado da entropia de  $Y$  condicionada aos eventos  $x$ . Observamos anteriormente (e será demonstrado posteriormente) que  $H(Y|X) \leq H(Y)$ , pois conhecer  $X$  reduz a incerteza sobre  $Y$ . Entretanto, devemos ressaltar que nada podemos dizer com relação a  $H(Y|X = x)$ , podendo este ser menor, igual ou maior que  $H(Y)$ . Para um caso específico, condicionar pode aumentar a incerteza, embora, na média, a entropia condicional seja sempre menor ou igual a entropia sem ser condicionada. Um exemplo interessante, adaptado de Cover e Thomas (2006), é o contexto de um julgamento: uma evidência única evidência pode aumentar a incerteza sobre um determinado caso, o que pode levar a um julgamento enviesado; entretanto, conhecer um conjunto de evidência nos leva esperar uma incerteza menor, eliminando o viés no julgamento.

Note que, condicionando  $H(Y|X)$  a  $Z$ , usando a Equação (33a) podemos escrever

$$H(Y|X, Z) = \sum_z p(z) H(Y|X, Z=z), \quad (34)$$

onde

$$H(Y|X, Z=z) = - \sum_{x,y} p(x,y|z) \log p(y|x,z). \quad (35)$$

Observando as definições de entropia conjunta e entropia condicional, surge, naturalmente, a constatação de que a entropia de um par de variáveis aleatórias é a entropia de uma somada à entropia condicional da outra. Esta observação é formalizada pelo teorema da regra da cadeia.

**Teorema 2** (Regra da Cadeia)

$$H(X, Y) = H(X) + H(Y|X) = H(Y) + H(X|Y) \quad (36)$$

*Demonstração.* Como  $p(x, y) = p(x)p(y|x)$  (teorema de Bayes), e como o logaritmo de um produto é a soma dos logaritmos dos fatores, teremos

$$-\log p(x, y) = -\log p(x) - \log p(y|x) \quad (37)$$

onde multiplicamos por  $(-1)$  ambos os lados. Calculando o valor esperado de ambos os lados, obtemos o resultado desejado.  $\square$

**Exemplo 1.** Considere duas variáveis aleatórias  $X$  e  $Y$ , com  $\mathcal{X} = \{x_1, x_2, x_3\}$  e  $Y = \{y_1, y_2, y_3\}$ . A distribuição conjunta  $p(x, y)$  é dada:

$p(x, y)$	$y_1$	$y_2$	$y_3$
$x_1$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
$x_2$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$
$x_3$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{4}$

A entropia conjunta é dada por

$$\begin{aligned} H(X, Y) &= -2\frac{1}{4}\log\frac{1}{4} - 2\frac{1}{8}\log\frac{1}{8} - 3\frac{1}{16}\log\frac{1}{16} - 2\frac{1}{32}\log\frac{1}{32} \\ &= 1 + \frac{6}{8} + \frac{3}{4} + \frac{10}{32} = 2.8125 \text{ bits.} \end{aligned}$$

Para calcular  $H(X)$  devemos primeiramente calcular a marginal

$$p(x) = \sum_y p(x, y) = \left\{ \frac{7}{16}, \frac{7}{32}, \frac{11}{32} \right\}.$$

e assim podemos calcular a entropia de  $X$

$$\begin{aligned} H(X) &= -\frac{7}{16}\log\frac{7}{16} - \frac{7}{32}\log\frac{7}{32} - \frac{11}{32}\log\frac{11}{32} \\ &= \frac{7}{16}(4 - \log 7) + \frac{7}{32}(5 - \log 7) + \frac{11}{32}(5 - \log 11) \\ &= \frac{146}{32} - \frac{21}{32}\log 7 + \frac{55}{32} - \frac{11}{32}\log 11 = 1.5310 \text{ bits.} \end{aligned}$$

Para calcular agora a entropia condicional  $H(Y|X)$ , devemos calcular a entropia condicional  $p(y|x)$

$p(y x)$	$y_1$	$y_2$	$y_3$
$x_1$	$\frac{4}{7}$	$\frac{4}{7}$	$\frac{2}{11}$
$x_2$	$\frac{2}{7}$	$\frac{2}{7}$	$\frac{1}{11}$
$x_3$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{8}{11}$

Dada as duas distribuições  $p(x, y)$  e  $p(y|x)$ , podemos agora utilizar a Equação (33c) para calcular  $H(Y|X)$ ,

$$\begin{aligned} H(Y|X) &= -\frac{1}{4}\log\frac{4}{7} - \frac{1}{8}\log\frac{4}{7} - \frac{1}{16}\log\frac{2}{11} - \frac{1}{8}\log\frac{2}{7} - \frac{1}{16}\log\frac{2}{7} - \frac{1}{32}\log\frac{1}{11} - \frac{1}{16}\log\frac{1}{7} - \frac{1}{32}\log\frac{1}{7} - \frac{1}{4}\log\frac{8}{11} \\ &= -\frac{3}{8}\log\frac{4}{7} - \frac{3}{16}\log\frac{2}{7} - \frac{3}{32}\log\frac{1}{7} - \frac{1}{16}\log\frac{2}{11} - \frac{1}{32}\log\frac{1}{11} - \frac{1}{4}\log\frac{8}{11} \\ &= \frac{3}{8}(\log 7 - 2) + \frac{3}{16}(\log 7 - 1) + \frac{3}{32}(\log 7) + \frac{1}{16}(\log 11 - 1) + \frac{1}{32}(\log 11) + \frac{1}{4}(\log 11 - 3) \\ &= \frac{21}{32}\log 7 + \frac{11}{32}\log 11 - \frac{56}{32} \\ &= 1.2815 \text{ bits.} \end{aligned}$$



Com os resultados anteriores podemos verificar a regra da cadeia:

$$H(X, Y) = H(X) + H(Y|X) \Rightarrow 2.8125 = 1.5308 + 1.2817 \text{ bits.}$$

Mesmo para um exemplo simples como este, é mais prático calcular numericamente, como apresentado no código Octave a seguir:

```
p_xy = [1/4, 1/8, 1/16; 1/8, 1/16, 1/32; 1/16, 1/32, 1/4];
H_XY = -sum(sum(p_xy .* log2(p_xy)));
p_y = sum(p_xy, 1);
p_y_given_x = p_xy ./ p_y; % p(y|x);
H_Y_given_X = -sum(sum(p_xy .* log2(p_y_given_x)));
```

### *Informação Mútua e Entropia Relativa*

Definimos anteriormente a entropia como uma medida da incerteza sobre uma variável aleatória, e a entropia condicional como a incerteza condicionada ao conhecimento de outra variável aleatória. Estas definições nos levam a questionar o quanto uma variável aleatória revela sobre a outra, ou, equivalentemente, o quanto a incerteza sobre uma é reduzida ao se conhecer a outra. Esta medida é dada pela informação mútua, definida como

**Definição 4** (Informação Mútua).

$$I(X; Y) \triangleq H(X) - H(X|Y) = H(Y) - H(Y|X), \quad (38)$$

onde as duas igualdades são válidas, por simetria. A primeira reflete a redução da incerteza sobre  $X$  quando  $Y$  é conhecido, e, de forma similar, a segunda reflete a redução da incerteza sobre  $Y$  quando  $X$  é conhecido.

Note que a notação  $I(X; Y)$  usa ponto e vírgula em vez de vírgula para enfatizar que a informação mútua mede a relação ou dependência entre as variáveis  $X$  e  $Y$ , distinguindo-se de uma função conjunta como  $H(X, Y)$ , onde a vírgula separa argumentos de uma distribuição.

A partir da definição dada na Equação (38), podemos reescrevê-la como a seguir:

$$I(X; Y) = H(X) - H(X|Y) \quad (39a)$$

$$= E_{p(x)} \log \frac{1}{p(x)} - E_{p(x,y)} \log \frac{1}{p(x|y)} \quad (39b)$$

$$= E_{p(x,y)} \log \frac{p(x|y)}{p(x)} \quad (39c)$$

$$= E_{p(x,y)} \log \frac{p(x|y)p(y)}{p(x)p(y)} \quad (39d)$$

$$= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \quad (39e)$$

A informação mútua, dada pela Equação (39e), pode ser vista como a divergência de Kullback–Leibler entre a distribuição conjunta e o produto das marginais, ou seja,

$$I(X; Y) = D(p(x, y) || p(x)p(y)). \quad (40)$$

Vejamos então a definição de divergência de Kullback–Leibler.

**Definição 5** (Divergência de Kullback–Leibler). A divergência de Kullback–Leibler (KL) entre duas distribuições  $p$  e  $q$ , em um alfabeto comum  $\mathcal{X}$ , é definida como

$$D(p || q) \triangleq \sum_x p(x) \log \frac{p(x)}{q(x)} \quad (41a)$$

$$= E_p \log \frac{p(X)}{q(X)}. \quad (41b)$$

Note que, em geral, a divergência de KL não é simétrica, ou seja,  $D(p || q) \neq D(q || p)$ .

Na definição de acima, adotamos a convenção de que o somatório se dá sobre o suporte da distribuição  $p$ ,  $S_p$ .

A divergência de KL é conhecida também na literatura como entropia relativa. Por não ser simétrica, não pode ser considerada como uma métrica ou distância, e ainda não satisfaz a desigualdade triangular.

Como observado anteriormente, seja  $\mu_1(x, y) = p(x, y)$  (distribuição conjunta) e  $\mu_2(x, y) = p(x)p(y)$  (produto das marginais), com  $p(x) = \sum_y p(x, y)$  e  $p(y) = \sum_x p(x, y)$ , então

$$D(\mu_1 || \mu_2) = \sum_{x,y} \mu_1(x, y) \log \frac{\mu_1(x, y)}{\mu_2(x, y)} \quad (42a)$$

$$= \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} = I(X; Y). \quad (42b)$$

A informação mútua é a distância entre a distribuição conjunta em  $X$  e  $Y$  e o produto das distribuições marginais em  $X$  e  $Y$ . Se as variáveis aleatórias são independentes, teremos  $p(x, y) = p(x)p(y)$  e por conseguinte a divergência será nula, a informação mútua entre  $X$  e  $Y$  será zero. A informação mútua é o erro em se assumir independência entre as variáveis aleatórias.

O produto das distribuições marginais  $p(x)p(y)$  é uma projeção da distribuição conjunta  $p(x, y)$  sobre o conjunto dos produtos de distribuições independentes, minimizando a divergência entre a distribuição conjunta e o produto das marginais. Ou seja,

$$p(x)p(y) = \underset{p'(x,y) \setminus p'(x,y)=p'(x)p'(y)}{\operatorname{argmin}} D(p(x, y) || p'(x, y)). \quad (43)$$

Podemos agora considerar uma aplicação prática em problemas de estimação paramétrica: minimizar a divergência KL entre uma distribuição subjacente  $p_\theta$  e uma distribuição empírica  $\hat{p}$  revela-se equivalente a maximizar o logaritmo da verossimilhança dos dados sob o modelo  $\hat{p}$ . A divergência KL quantifica o custo ao usar  $\hat{p}$  para aproximar  $p_\theta$ , e otimizar esse custo nos leva diretamente a ajustar os parâmetros  $\theta$  para melhor descrever os dados observados.

Seja  $x_1, \dots, x_N \in \mathcal{X}$ ,  $N$  observações i.i.d.<sup>8</sup> de uma variável aleatória  $X$ . A distribuição empírica será dada por

$$\hat{p}(x) = \frac{1}{N} \sum_{n=1}^N \delta(x - x_n), \quad (44)$$

onde  $\delta$  é a função de Dirac.

Seja  $p_\theta$  uma distribuição em  $\mathcal{X}$  parametrizada por  $\theta$ . Maximizar a verossimilhança de  $p_\theta(x)$  é equivalente a minimizar a divergência de KL  $D_{\text{KL}}(\hat{p} \parallel p_\theta)$ .

$$D_{\text{KL}}(\hat{p} \parallel p_\theta) = \sum_{x \in \mathcal{X}} \hat{p}(x) \log \frac{\hat{p}(x)}{p_\theta(x)} \quad (45a)$$

$$= -H(\hat{p}) - \sum_{x \in \mathcal{X}} \hat{p}(x) \log p_\theta(x) \quad (45b)$$

$$= -H(\hat{p}) - \frac{1}{N} \sum_{x \in \mathcal{X}} \sum_{n=1}^N \delta(x - x_n) \log p_\theta(x) \quad (45c)$$

$$= -H(\hat{p}) - \frac{1}{N} \sum_{n=1}^N \log p_\theta(x_n) \quad (45d)$$

$$= -H(\hat{p}) - \ell(p_\theta(x)), \quad (45e)$$

onde  $\ell(\cdot)$  é a função log-verossimilhança.

A estimativa de máxima verossimilhança de  $\theta$  a partir das  $N$  observações é dada por

$$\hat{\theta}_n = \operatorname{argmax}_{\theta \in \Theta} \prod_{n=1}^N p_\theta(x_n) \quad (46a)$$

$$= \operatorname{argmax}_{\theta \in \Theta} \sum_{n=1}^N \log p_\theta(x_n) \quad (46b)$$

$$= \operatorname{argmin}_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^N -\log p_\theta(x_n). \quad (46c)$$

Desta forma, podemos constatar que a distribuição que minimiza a divergência de KL para a distribuição empírica é aquela que maximiza a verossimilhança (ou logaritmo desta). Pela lei forte dos grandes números<sup>9</sup>,  $\frac{1}{N} \sum_{i=1}^N \log p_\theta(x_n) \xrightarrow{q.c.} E[\log p_\theta(X)]$ .

Ao ajustar os parâmetros  $\theta$  para minimizar divergência KL, estamos buscando o modelo  $\hat{p}$  que melhor representa  $p$ . A não negatividade da

<sup>8</sup> Uma coleção de variáveis aleatórias é independente e identicamente distribuída (i.i.d.) se todas possuírem uma mesma distribuição e forem mutuamente independentes.

<sup>9</sup> A Lei Forte dos Grandes Números afirma que, dada um sequência infinita  $X_1, X_2, X_3, \dots$  de variáveis aleatórias independentes e identicamente distribuídas (i.i.d.) com média finita  $E[X_i] = \mu$ , a média amostral  $\bar{X}_n = 1/n \sum_{i=1}^n X_i$  converge quase certamente para  $\mu$ , ou seja,  $\bar{X}_n \xrightarrow{q.c.} \mu$ .

divergência, garante que o custo informacional nunca será negativo e que a aproximação ótima ocorre quando as distribuições coincidem. Graças à convexidade da função logarítmica, sabemos que esse ótimo é bem definido e único.”

Vejamos então uma propriedade essencial da divergência: a não negatividade.

**Teorema 3** (Não negatividade da Divergência) *Para duas distribuições  $p$  e  $q$  em um alfabeto comum  $\mathcal{X}$ ,*

$$D(p||q) \geq 0 \quad (47)$$

*com igualdade se e somente se  $p = q$ .*

*Demonstração.* Se  $q(x) = 0$  para algum  $x \in S_p$ , então  $D(p||q) = \infty$  e o teorema é verdadeiro (caso trivial). Vamos assumir então que  $q(x) > 0 \forall x \in S_p$ . Teremos então

$$D(p||q) = \log e \sum_{x \in S_p} p(x) \ln \frac{p(x)}{q(x)} \quad (48a)$$

$$\geq \log e \sum_{x \in S_p} p(x) \left( 1 - \frac{q(x)}{p(x)} \right) \quad (48b)$$

$$= \log e \left[ \sum_{x \in S_p} p(x) - \sum_{x \in S_p} q(x) \right] \quad (48c)$$

$$\geq \log e(1 - 1) \quad (48d)$$

$$= 0, \quad (48e)$$

onde em 48b utilizamos a Equação (28) substituindo  $z$  por  $1/z$  (o que fornece  $\ln z \geq 1 - 1/z$ ) e para obter 48d utilizamos o fato de que

$$\sum_{x \in S_p} q(x) \leq 1, \quad (49)$$

uma vez que o somatório é tomado no suporte de  $p$ , podendo assim excluir alguns pontos não nulos de  $q$ .

□

O fato da divergência ser não negativa, nos leva como consequência em termos também a não negatividade da informação mútua, uma vez que esta é um caso específico de divergência.

**Lema 4** (Não negatividade da Informação Mútua) *A informação mútua entre duas variáveis aleatórias  $X$  e  $Y$  é tal que*

$$I(X; Y) \geq 0, \quad (50)$$

*com igualdade se e somente se  $X \perp\!\!\!\perp Y$*

*Demonstração.* A informação mútua é a divergência entre a distribuição conjunta e o produto das marginais, logo, aplicando o Teorema 3, obtemos

$$I(X; Y) = D(p(x, y) || p(x)p(y)) \geq 0. \quad (51)$$

Quando  $X \perp Y$ , teremos  $p(x, y) = p(x)p(y)$  e assim a divergência será nula e também a informação mútua.  $\square$

A divergência também pode ser utilizada para demonstrar o limite máximo da entropia, de forma alternativa àquela apresentada na Equação (30).

**Teorema 5** (Limite Máximo da Entropia)  $H(X) \leq \log |\mathcal{X}|$ , onde  $|\mathcal{X}|$  denota a cardinalidade do alfabeto  $\mathcal{X}$ , com igualdade se e somente se  $X$  possuir distribuição uniforme.

*Demonstração.* Seja  $u(x) = 1/|\mathcal{X}|$  a função probabilidade de massa uniforme em  $\mathcal{X}$ , e seja  $p(x)$  a função probabilidade de massa para  $X$ . Então

$$\begin{aligned} D(p||u) &= \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{u(x)} \\ &= \sum_{x \in \mathcal{X}} p(x) \log p(x) + \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{u(x)} \\ &= -H(X) + \log |\mathcal{X}| \sum_{x \in \mathcal{X}} p(x) \\ &= \log |\mathcal{X}| - H(X). \end{aligned} \quad (52)$$

Como a entropia relativa é não negativa,  $D(p||u) \geq 0$ , teremos

$$D(p||u) = \log |\mathcal{X}| - H(X) \geq 0, \quad (53)$$

e assim

$$H(X) \leq \log |\mathcal{X}| \quad (54)$$

$\square$

**Observação 1** Para uma variável aleatória em um alfabeto  $D$ -ário, ou seja,  $|\mathcal{X}| = D$ , a entropia na base  $D$  será

$$H_D(X) \leq 1. \quad (55)$$

**Observação 2** O Teorema 5 estabelece um valor máximo para a entropia desde que o alfabeto seja finito. Para o caso de alfabeto de tamanho infinito, a entropia pode ser finita ou não.

Vejamos dois exemplos:

**Exemplo 2.** Seja  $X$  uma variável aleatória com distribuição dada por

$$Pr(X = i) = 2^{-i}, \quad i = 1, 2, \dots \quad (56)$$

Neste caso, teremos

$$H(X) = \sum_{i=1}^{\infty} i 2^{-i} = 2. \quad (57)$$

**Exemplo 3** (Yeung 2002). Considere uma variável aleatória  $X$  com valores em pares de inteiros

$$\left\{ (i, j) : 1 \leq i \leq \infty \text{ e } 1 \leq j \leq \frac{2^{2^i}}{2^i} \right\} \quad (58)$$

tal que

$$Pr(X = (i, j)) = 2^{-2^i}, \quad (59)$$

para todo  $i$  e  $j$ . A entropia será dada por

$$H(X) = - \sum_{i=1}^{\infty} \sum_{j=1}^{2^{2^i}/2^i} 2^{-2^i} \log 2^{-2^i} = \sum_{i=1}^{\infty} 1 \quad (60)$$

que não converge.

Vejamos ainda algumas outras observações que podemos fazer sobre a informação mútua. Primeiramente, pela Definição 4, é fácil notar que a informação mútua  $I(X; Y)$  é simétrica em  $X$  e  $Y$ , ou seja,  $I(X; Y) = I(Y; X)$ .

Em seguida, vamos analisar o que ocorre quando tomamos a informação mútua entre uma variável aleatória  $X$  e ela mesma, ou seja,  $I(X; X)$ .

**Proposição 1** (A informação mútua de uma variável aleatória e ela mesma é a entropia) *A informação mútua de uma variável aleatória  $X$  com ela mesma, a auto-informação de  $X$ , é igual à entropia de  $X$ .*

*Demonstração.*

$$I(X; X) = E \log \frac{p(X)}{p(X)^2} \quad (61a)$$

$$= -E \log p(X) \quad (61b)$$

$$= H(X). \quad (61c)$$

□

Podemos descrever uma nova relação, ao observar a Definição 4 ( $I(X; Y) = H(X) - H(X|Y)$ ) e o Teorema 2 ( $H(X|Y) = H(X, Y) - H(Y)$ )

**Proposição 2**

$$I(X; Y) = H(X) + H(Y) - H(X, Y). \quad (62)$$

Todas essas relações observadas entre entropia, entropia conjunta e informação mútua podem ser facilmente representadas através do diagrama de Venn apresentado na Figura 8. Esta correspondência entre o diagrama de Venn e as medidas de informação de Shannon não são mera coincidência.

De forma semelhante que vimos anteriormente, quando foi introduzido o conceito de entropia condicional, podemos agora estender o conceito de informação mútua entre duas variáveis aleatórias  $X$  e  $Y$  dado o conhecimento de uma terceira variável  $Z$ , definindo assim a informação mútua condicional.

**Definição 6** (Informação Mútua Condicional). Dadas as variáveis aleatórias  $X$ ,  $Y$  e  $Z$ , a informação mútua entre  $X$  e  $Y$ , condicionada ao conhecimento de  $Z$  é dada por

$$I(X; Y|Z) \triangleq \sum_z p(z) I(X; Y|Z = z) \quad (63a)$$

$$= \sum_z p(z) E_{p(x,y,z)} \log \frac{p(x, y|Z = z)}{p(x|Z = z)p(y|Z = z)} \quad (63b)$$

$$= \sum_{x,y,z} p(x, y, z) \log \frac{p(x, y|z)}{p(x|z)p(y|z)} \quad (63c)$$

$$= \sum_{x,y,z} p(x, y, z) \log \frac{p(x|y, z)p(y|z)}{p(x|z)p(y|z)} \quad (63d)$$

$$= E_{p(x,y,z)} \left[ \log \frac{1}{p(x|z)} - \log \frac{1}{p(x|y, z)} \right] \quad (63e)$$

$$= H(X|Z) - H(X|Y, Z). \quad (63f)$$

*Generalização da Regra da Cadeia*

A generalização é fundamental para analisar sistemas com múltiplas variáveis ou séries temporais. Uma sequência de variáveis aleatórias  $X_1, X_2, \dots, X_N = X_{1:N}$  pode representar estados ou observações ao longo do tempo. A regra da cadeia, vista anteriormente no Teorema 2, pode ser generalizada para uma sequência de  $N$  variáveis aleatórias  $X_{1:N}$ .

**Proposição 3** (Regra da Cadeia Generalizada da Entropia) *Dada uma sequência de  $N$  variáveis aleatórias  $X_{1:N}$ , a entropia conjunta pode ser dada por*

$$H(X_1, X_2, \dots, X_N) = \sum_{i=1}^N H(X_i|X_1, \dots, X_{i-1}). \quad (64)$$

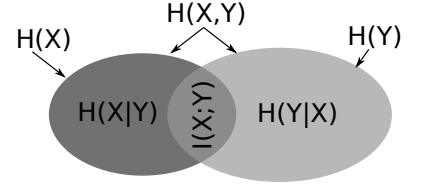


Figura 8: Diagrama de Venn ilustrando as relações entre entropias, entropias condicionais e informação mútua, facilitando a visualização das grandezas informacionais.

*Demonstração.* A demonstração é feita por indução. Sabemos que para  $N = 2$  é verdadeiro, como visto no Teorema 2. Vamos supor que para  $N = M$  é verdadeiro e mostrar que para  $N = M + 1$  é verdadeiro.

$$H(X_1, X_2, \dots, X_M, X_{M+1}) = H(X_1, X_2, \dots, X_M) + H(X_{M+1}|X_1, X_2, \dots, X_M) \quad (65a)$$

$$= \sum_{i=1}^M H(X_i|X_1, \dots, X_{i-1}) + H(X_{M+1}|X_1, X_2, \dots, X_M) \quad (65b)$$

$$= \sum_{i=1}^{M+1} H(X_i|X_1, \dots, X_{i-1}), \quad (65c)$$

onde em 65a utilizamos o Teorema 2, fazendo  $X = (X_1, X_2, \dots, X_M)$  e  $Y = X_{M+1}$ , e em 65b utilizamos a premissa de que a relação 64 é válida para  $N = M$ . Isto prova então que a relação 64 é válida para  $N = M + 1$  e, como é sabidamente válida para  $N = 2$ , será válida para todo  $N$ . □

A regra da cadeia vista no Teorema 2 pode ser estendida para o caso do condicionamento à uma terceira variável aleatória.

**Proposição 4** (Regra da Cadeia Condicional) *Para duas variáveis aleatórias  $X$  e  $Y$  condicionadas a uma terceira,  $Z$ , a entropia conjunta condicional pode ser decomposta como:*

$$H(X, Y|Z) = H(X|Z) + H(Y|X, Z). \quad (66)$$

*A incerteza total sobre  $(X, Y)$ , dado  $Z$ , pode ser quebrada em duas partes: primeiro, a incerteza sobre  $X$  dado  $Z$ , e depois a incerteza adicional de  $Y$  dado tanto  $X$  quanto  $Z$ .*

*Demonstração.* A demonstração segue os mesmos passos da demonstração do Teorema 2, bastando condicionar a  $Z$  para obtermos a adaptação para a regra da cadeia condicional. □

Fazemos agora a generalização para uma sequência de variáveis aleatórias  $X_{1:N}$  condicionada a uma variável aleatória  $Y$ .

**Proposição 5** (Regra da Cadeia Condicional Generalizada)

$$H(X_1, X_2, \dots, X_N|Y) = \sum_{i=1}^N H(X_i|X_1, \dots, X_{i-1}, Y). \quad (67)$$



*Demonstração.*

$$H(X_1, X_2, \dots, X_N|Y) = \sum_y H(X_1, X_2, \dots, X_N|Y = y) \quad (68a)$$

$$= \sum_y p(y) \sum_{i=1}^N H(X_i|X_1, \dots, X_{i-1}, Y = y) \quad (68b)$$

$$= \sum_{i=1}^N \sum_y p(y) H(X_i|X_1, \dots, X_{i-1}, Y = y) \quad (68c)$$

$$= \sum_{i=1}^N H(X_i|X_1, \dots, X_{i-1}, Y), \quad (68d)$$

onde utilizamos 33a em 68a, 34 em 68d, e 68b segue de 64.

□

Por fim, podemos aplicar as mesmas ideias para obter a regra da cadeia para informação mútua, que descreve como a informação compartilhada entre um conjunto de variáveis e outra variável pode ser decomposta em contribuições condicionais.

**Proposição 6** (Regra da Cadeia da Informação Mútua)

$$I(X_1, X_2, \dots, X_N; Y) = \sum_{i=1}^N I(X_i; Y|X_1, \dots, X_{i-1}). \quad (69)$$

*Demonstração.*

$$I(X_1, X_2, \dots, X_N; Y) = H(X_1, X_2, \dots, X_N) - H(X_1, X_2, \dots, X_N|Y) \quad (70a)$$

$$= \sum_{i=1}^N [H(X_i|X_1, \dots, X_{i-1}) - H(X_i|X_1, \dots, X_{i-1}, Y)] \quad (70b)$$

$$= \sum_{i=1}^N I(X_i; Y|X_1, \dots, X_{i-1}), \quad (70c)$$

onde utilizamos as Proposições 3 e 5.

□

### *Desigualdade de Jensen*

Nesta seção, apresentamos a desigualdade de Jensen, uma relação observada para funções convexas entre o valor esperado da função de uma variável aleatória e a função do valor esperado desta variável aleatória. Mais precisamente,  $Ef(X) \geq f(EX)$ , para  $f$  uma função convexa e  $X$  uma variável aleatória.

A desigualdade de Jensen é uma ferramenta fundamental para demonstrar diversas propriedades essenciais. Entretanto, primeiramente devemos rever o teste da segunda derivada para convexidade.

**Definição 7** (Função Convexa). Dizemos que  $f$  é convexa em  $(a, b)$  se para todo  $x_1, x_2 \in (a, b)$ ,  $0 \leq \lambda \leq 1$ ,

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) \quad (71)$$

A Figura 9 ilustra uma função convexa no intervalo, evidenciando como a imagem de um ponto intermediário  $\lambda x_1 + (1 - \lambda)x_2$ , entre  $x_1$  e  $x_2$ , é menor ou igual ao correspondente na corda que passa por  $(x_1, f(x_1))$  e  $(x_2, f(x_2))$ .

Alguns exemplos de funções convexas são:  $f(x) = x^2$ ;  $f(x) = x^4$ ;  $f(x) = e^x$ ; e  $x \log x$ ,  $x \geq 0$ . Teremos que  $f$  é estritamente convexa se a igualdade for verdadeira apenas para  $\lambda = 0$  ou  $\lambda = 1$ . Uma função  $f$  é dita côncava se  $-f$  ( $f$  multiplicada por  $-1$ ) for uma função convexa, ou seja, côncavo é o oposto de convexo.

Para determinar se uma função é convexa, podemos realizar o simples teste da segunda derivada.

**Proposição 7** (Teste da Segunda Derivada para Convexidade) *Se uma função  $f$  possui derivada segunda não-negativa (positiva) em um intervalo, a função é convexa (estritamente convexa) no intervalo.*

*Demonstração.* A expansão de Taylor de uma função  $f$  em torno do ponto  $x_0$  é dada por

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x^*)}{2}(x - x_0)^2 \quad (72)$$

onde  $x^* \in (x_0, x)$ . Por hipótese,  $f''(x^*) \geq 0$ , e desta forma, o último termo é não-negativo.

Seja  $x_0 = \lambda x_1 + (1 - \lambda)x_2$ . Analisando em  $x = x_1$ , teremos

$$f(x_1) \geq f(x_0) + f'(x_0)(x_1 - \lambda x_1 - (1 - \lambda)x_2) \quad (73a)$$

$$= f(x_0) + f'(x_0)((1 - \lambda)(x_1 - x_2)) \quad (73b)$$

Da mesma forma, em  $x = x_2$ , teremos

$$f(x_2) \geq f(x_0) + f'(x_0)(x_2 - \lambda x_1 - (1 - \lambda)x_2) \quad (74a)$$

$$= f(x_0) + f'(x_0)(\lambda(x_2 - x_1)) \quad (74b)$$

Somando  $\lambda$  73b com  $(1 - \lambda)$  74b, obtemos

$$\lambda f(x_1) + (1 - \lambda)f(x_2) \geq \lambda f(x_0) + \lambda f'(x_0)((1 - \lambda)(x_1 - x_2)) + (1 - \lambda)f(x_0) + (1 - \lambda)f'(x_0)(\lambda(x_2 - x_1)) \quad (75a)$$

$$\geq f(x_0) = f(\lambda x_1 + (1 - \lambda)x_2) \quad (75b)$$

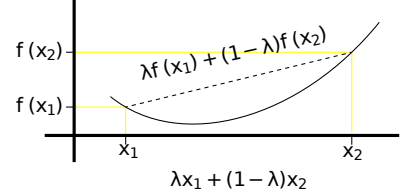


Figura 9: Ilustração de uma função convexa no intervalo.

□

Podemos agora retornar à desigualdade de Jensen. Vemos na Figura 10 uma ilustração do que tal propriedade representar.

**Teorema 6** (Desigualdade de Jensen) *Seja  $f$  uma função convexa e  $X$  uma variável aleatória, então*

$$E[f(X)] = \sum_x p(x)f(x) \geq f(EX) = f\left(\sum_x xp(x)\right) \quad (76)$$

*Demonstração.* Para uma distribuição de massa com apenas dois pontos

$$E[f(X)] = p_1 f(x_1) + p_2 f(x_2) \geq f(p_1 x_1 + p_2 x_2) = f(EX) \quad (77)$$

já que  $f$  é convexa e  $p_1 + p_2 = 1$ .

Para uma distribuição com mais de dois pontos, iremos fazer uma demonstração por indução.

Suponha que o teorema seja verdadeiro para uma distribuição com  $k - 1$  pontos de massa. Para uma distribuição com  $k$  pontos de massa podemos escrever cada  $p'_i = p_i / (1 - p_k)$  para  $i = 1, 2, \dots, k - 1$ .

Desta forma, teremos

$$E[f(X)] = \sum_{i=1}^k p_i f(x_i) \quad (78a)$$

$$= \sum_{i=1}^{k-1} (1 - p_k) p'_i f(x_i) + p_k f(x_k) \quad (78b)$$

$$= (1 - p_k) \sum_{i=1}^{k-1} p'_i f(x_i) + p_k f(x_k) \quad (78c)$$

$$\geq (1 - p_k) f\left(\sum_{i=1}^{k-1} p'_i x_i\right) + p_k f(x_k) \quad (78d)$$

$$\geq f\left((1 - p_k) \sum_{i=1}^{k-1} p'_i x_i + p_k x_k\right) \quad (78e)$$

$$= f\left(\sum_{i=1}^k p_i x_i\right) \quad (78f)$$

onde em 78c utilizamos a hipótese de indução, já que

$$\sum_{i=1}^{k-1} p'_i = \sum_{i=1}^{k-1} \frac{p_i}{1 - p_k} = \frac{1 - p_k}{1 - p_k} = 1. \quad (79)$$

e em 78d utilizamos a definição de convexidade.

Desta forma, sendo o teorema válido para uma distribuição de massa com  $k - 1$  pontos, também será verdadeiro para uma distribuição de massa com  $k$  pontos. Como mostramos que para  $k = 2$  é verdadeiro, logo o teorema é verdadeiro para qualquer  $k$ .

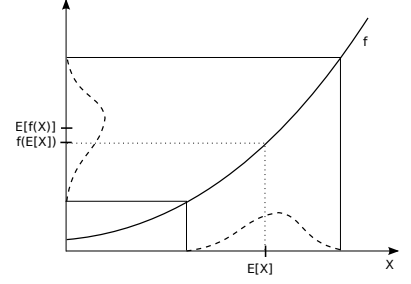


Figura 10: Ilustração da desigualdade de Jensen, fornecendo uma intuição para ela.

□

A desigualdade de Jensen pode ser utilizada para demonstrar a não negatividade da divergência de KL.

**Lema 7** (Não negatividade da Divergência de Kullback–Leibler)

$$D(p||q) \geq 0 \text{ com igualdade se e somente se } p(x) = q(x) \forall x. \quad (80)$$

*Demonstração.* De forma equivalente, iremos mostrar que  $-D(p||q) \leq 0$ .

Seja  $S_p = \{x : p(x) > 0\}$ , o suporte de  $p$ , então

$$-D(p||q) = -\sum_x p(x) \log \frac{p(x)}{q(x)} = -\sum_{x \in S_p} p(x) \log \frac{p(x)}{q(x)} \quad (81a)$$

$$= \sum_{x \in S_p} p(x) \log \frac{q(x)}{p(x)} = E \log \frac{q(X)}{p(X)} \quad (81b)$$

$$\leq \log \left( E \frac{q(X)}{p(X)} \right) = \log \left( \sum_{x \in S_p} p(x) \frac{q(x)}{p(x)} \right) \quad (81c)$$

$$= \log \left( \sum_{x \in S_p} q(x) \right) \leq \log \left( \sum_x q(x) \right) = \log 1 = 0 \quad (81d)$$

onde em 81b utilizamos a desigualdade de Jensen, Equação (76).  $\square$

A desigualdade da soma dos logaritmos é derivada da desigualdade de Jensen e da convexidade da função logarítmica. Esta desigualdade também aparece em algumas demonstrações importantes.

**Proposição 8** (Desigualdade da Soma de Logaritmos) *Dados  $(a_1, \dots, a_n)$  e  $(b_1, \dots, b_n)$ , com  $a_i \geq 0$  e  $b_i \geq 0$ , temos*

$$\sum_{i=1}^n a_i \log \frac{a_i}{b_i} \geq \left( \sum_{i=1}^n a_i \right) \log \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} \quad (82)$$

*e teremos igualdade se e somente se  $a_i/b_i = c$ , onde  $c$  é uma constante.*

*Demonstração.* Considere  $f(t) = t \log t = t(\ln t)(\log e)$ , que é estritamente convexa, pois  $f''(t) = 1/t \log e > 0$ ,  $\forall t > 0$ .

Dada  $f$  convexa, a desigualdade de Jensen diz que

$$\sum_i \alpha_i f(t_i) \geq f \left( \sum_i \alpha_i t_i \right) \text{ com } \alpha_i \geq 0 \text{ e } \sum_i \alpha_i = 1. \quad (83)$$

$f(x) = x \log x$  é estritamente convexa para  $x > 0$ , já que  $f''(x) = \frac{1}{x} \log e > 0$  para  $x > 0$ .

Vamos fazer  $\alpha_i = b_i / \sum_{j=1}^n b_j$  e  $t_i = a_i / b_i$ , então obteremos

$$\sum_i \alpha_i f(t_i) \geq f\left(\sum_i \alpha_i t_i\right) \quad (84a)$$

$$\sum_i \left( \frac{b_i}{\sum_j b_j} f\left(\frac{a_i}{b_i}\right) \right) \geq f\left(\sum_i \frac{b_i}{\sum_j b_j} \frac{a_i}{b_i}\right) \quad (84b)$$

$$\frac{1}{\sum_j b_j} \sum_i \left( b_i \frac{a_i}{b_i} \log \frac{a_i}{b_i} \right) \geq \left( \sum_i \frac{a_i}{\sum_j b_j} \right) \log \sum_i \frac{a_i}{\sum_j b_j} \quad (84c)$$

$$\sum_i a_i \log \frac{a_i}{b_i} \geq \left( \sum_i a_i \right) \log \sum_i \frac{a_i}{\sum_j b_j} \quad (84d)$$

$$\sum_i a_i \log \frac{a_i}{b_i} \geq \sum_i a_i \log \frac{\sum_i a_i}{\sum_j b_j}. \quad (84e)$$

□

A desigualdade da soma de logaritmos pode ser utilizada para mostrar que  $D(p||q) \geq 0$ .

**Proposição 9** (Não negatividade da Divergência)  $D(p||q) \geq 0$

*Demonstração.*

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \quad (85a)$$

$$\geq \left( \sum_x p(x) \right) \log \frac{\sum_x p(x)}{\sum_x q(x)} \quad (85b)$$

$$= 1 \log \frac{1}{1} = 0. \quad (85c)$$

□

A entropia relativa, ou divergência de Kullback-Leibler, possui uma propriedade importante: ela é convexa no par de distribuições. A convexidade no par considera a interpolação simultânea de ambas as distribuições e garante que a divergência KL se comporte de forma previsível ao combinar distribuições.

**Teorema 8** (A Entropia Relativa é Convexa no Par) *Para dois pares de distribuições  $(p_1, q_1)$  e  $(p_2, q_2)$ ,*

$$D(\lambda p_1 + (1-\lambda)p_2 || \lambda q_1 + (1-\lambda)q_2) \leq \lambda D(p_1 || q_1) + (1-\lambda)D(p_2 || q_2), \quad (86)$$

para todo  $0 \leq \lambda \leq 1$ .

*Demonstração.* Usando a Definição 5, definição de divergência de KL, temos

$$D(\lambda p_1 + (1 - \lambda)p_2 || \lambda q_1 + (1 - \lambda)q_2) = \sum_x (\lambda p_1(x) + (1 - \lambda)p_2(x)) \log \frac{\lambda p_1(x) + (1 - \lambda)p_2(x)}{\lambda q_1(x) + (1 - \lambda)q_2(x)} \quad (87a)$$

$$\leq \sum_x \left( \lambda p_1(x) \log \frac{\lambda p_1(x)}{\lambda q_1(x)} + (1 - \lambda)p_2(x) \log \frac{(1 - \lambda)p_2(x)}{(1 - \lambda)q_2(x)} \right) \quad (87b)$$

$$= \lambda D(p_1 || q_1) + (1 - \lambda)D(p_2 || q_2) \quad (87c)$$

onde, em 87a, podemos considerar que, dentro do somatório em  $x$ , temos um somatório com dois termos:  $(a_1 + a_2) \log ((a_1 + a_2)/(b_1 + b_2))$ . Utilizamos então a Equação (82) para o caso com apenas dois termos:

$$\left( \sum_{i=1}^2 a_i \right) \log \frac{\sum_{i=1}^2 a_i}{\sum_{i=1}^2 b_i} \leq \sum_{i=1}^2 a_i \log \frac{a_i}{b_i} = a_1 \log \frac{a_1}{b_1} + a_2 \log \frac{a_2}{b_2}, \quad (88)$$

o que nos leva ao resultado em 87b. E usamos novamente a Definição 5 para obter 87c. □

**Teorema 9** (A Entropia Relativa é Convexa no Primeiro Argumento)

*A divergência de Kullback-Leibler,  $D(p||q)$ , é convexa em relação a  $p$ , fixando-se  $q$ . Ou seja, dadas  $p_1$  e  $p_2$ , duas distribuições de probabilidade, e fixando-se  $q$ , a distribuição de referência ou distribuição base. Temos então, para  $\lambda \in [0, 1]$ ,*

$$D(\lambda p_1 + (1 - \lambda)p_2 | q) \leq D(p_1 | q) + (1 - \lambda)D(p_2 | q). \quad (89)$$

*Demonstração.*

$$D(\lambda p_1 + (1 - \lambda)p_2 | q) = \sum_x (\lambda p_1 + (1 - \lambda)p_2) \log \frac{\lambda p_1 + (1 - \lambda)p_2}{q} \quad (90a)$$

$$\leq \lambda \sum_x p_1 \log \frac{p_1}{q} + (1 - \lambda) \sum_x p_2 \log \frac{p_2}{q} \quad (90b)$$

$$= \lambda D(p_1 || q) + (1 - \lambda)D(p_2 || q), \quad (90c)$$

onde em 90a utilizamos que a função  $f(p) = p \log p$  é convexa em  $p$  (lembrando que  $q$  é fixo e assim  $f''(p) = 1/p > 0$ , para  $p > 0$ ) e utilizando a desigualdade de Jensen (Teorema 6). Em 90b apenas utilizamos a definição de divergência (Definição 5). □

A entropia relativa, entretanto, não é convexa no segundo argumento. Será entretanto côncava em  $q$  para  $p$  fixo. A demonstração deixamos a cargo do leitor.

O Teorema 9 pode ser utilizado para demonstrar a concavidade da entropia.

**Teorema 10** (A Entropia é Côncava)  $H(p)$  é uma função concava de  $p$ .

*Demonstração.*

$$H(p) = - \sum_i p_i \log p_i = - \sum_i p_i \log p_i + \log |\mathcal{X}| - \log |\mathcal{X}| \quad (91a)$$

$$= \log |\mathcal{X}| - \sum_i p_i \log p_i - \log |\mathcal{X}| \underbrace{\sum_i p_i}_{=1} \quad (91b)$$

$$= \log |\mathcal{X}| - \sum_i (p_i \log p_i + p_i \log |\mathcal{X}|) \quad (91c)$$

$$= \log |\mathcal{X}| - \sum_i p_i (\log p_i - \log 1/|\mathcal{X}|) \quad (91d)$$

$$= \underbrace{\log |\mathcal{X}|}_{\text{constante}} - \underbrace{D(p||u)}_{\text{convexo}} \quad (91e)$$

□

A partir da relação em 91e, podemos ver a entropia como a similaridade com a distribuição uniforme. Quanto maior a entropia (menor a divergência  $D(p||u)$ ), mais próximo estaremos da distribuição uniforme.

Analisaremos agora a convexidade ou concavidade da informação mútua  $I(X; Y)$ , quando fixamos uma das distribuições condicionais ou marginais. Essa análise possui implicação direta na determinação da capacidade de canal em teoria da comunicação, onde a maximização da informação mútua, sobre a distribuição de entrada, será um problema bem comportado com um único máximo global. A concavidade da função objetivo implica que o máximo é atingível e pode ser encontrado usando técnicas de otimização convexa.

**Teorema 11** (Concavidade/Convexidade da Informação Mútua) *Seja  $(X, Y) \sim p(x, y) = p(x)p(y|x)$ , a informação mútua  $I(X; Y)$  é uma função côncava de  $p(x)$  para  $p(y|x)$  fixo e uma função convexa de  $p(y|x)$  para  $p(x)$  fixo.*

*Demonstração.* Vejamos primeiro o motivo de  $I(X; Y)$  ser uma função côncava de  $p(x)$  para  $p(y|x)$  fixo. Podemos escrever a informação

mútua como uma função de  $p(x)$ :

$$I(X; Y) = D(p(x, y) || p(x)p(y)) \quad (92a)$$

$$= \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (92b)$$

$$= \sum_{x,y} p(x)p(y|x) \log \frac{p(x)p(y|x)}{p(x) \sum_x p(x)p(y|x)}, \quad (92c)$$

onde em 92a utilizamos a definição de informação mútua, em 92b utilizamos a definição de divergência e aplicamos o teorema de Bayes em 92c. Se  $p(y|x)$  é constante, então a informação mútua é função de  $p(x)$ :  $I_{p(x)}(X; Y)$ . Utilizando a propriedade da convexidade da divergência de Kullback-Leibler,

$$I_{\lambda p_1(x) + (1-\lambda)p_2(x)}(X; Y) \geq \lambda I_{p_1(x)}(X; Y) + (1-\lambda) I_{p_2(x)}(X; Y). \quad (93)$$

Então a informação mútua é uma função concava de  $p(x)$  para  $p(y|x)$  fixo.

Agora iremos demonstrar que  $I(X; Y)$  é uma função convexa de  $p(y|x)$  para  $p(x)$  fixo. Aplicamos a mesma ideia, porém agora consideraremos  $p(x)$  fixo. Escrevemos então a informação mútua como uma função de  $p(y|x)$ :

$$I_{p(y|x)}(X; Y) = \sum_{x,y} p(x)p(y|x) \log \frac{p(x)p(y|x)}{p(x) \sum_x p(x)p(y|x)}. \quad (94)$$

Utilizando a propriedade da convexidade da divergência de Kullback-Leibler, obtemos

$$I_{\lambda p_1(y|x) + (1-\lambda)p_2(y|x)}(X; Y) \leq \lambda I_{p_1(y|x)}(X; Y) + (1-\lambda) I_{p_2(y|x)}(X; Y). \quad (95)$$

□

### *Desigualdade de Processamento de Dados*

Quando falamos em processamento de dados, podemos pensar no seguinte modelo:  $X \rightarrow Y \rightarrow Z$ , onde a variável aleatória  $X$  representa os dados originais, a serem transmitidos;  $Y$  representa os dados recebidos no outro lado de um canal de comunicação; e  $Z$  o resultado de um processamento adicional aplicado a  $Y$ . O processamento de dados é modelado como uma cadeia de Markov. Nessa estrutura,  $Z$  depende de  $X$  apenas através de  $Y$ , ou, de outra forma,  $Z$  e  $X$  são condicionalmente independentes, dado  $Y$ .



**Definição 8** (Cadeia de Markov). As variáveis aleatórias  $X$ ,  $Y$  e  $Z$  formam uma cadeia de Markov nesta ordem (denotado  $X \rightarrow Y \rightarrow Z$ ) se a distribuição condicional de  $Z$  depende apenas de  $Y$  e é condicionalmente independente de  $X$ . Especificamente,  $X$ ,  $Y$  e  $Z$  formam uma cadeia de Markov  $X \rightarrow Y \rightarrow Z$  se a função massa de probabilidade conjunta pode ser escrita como

$$p(x, y, z) = p(x)p(y|x)p(z|y) \quad (96)$$

Temos uma cadeia de Markov  $X \rightarrow Y \rightarrow Z$  se, e somente se,  $X$  e  $Z$  são condicionalmente independentes dado  $Y$  ( $X \perp\!\!\!\perp Z|Y$ ). Isto é,

$$p(x, z|y) = \frac{p(x, y, z)}{p(y)} = \frac{p(x)p(y|x)p(z|y)}{p(y)} = p(x|y)p(z|y) \quad \forall x, y, z \quad (97)$$

**Proposição 10** (Cadeia de Markov Inversa) *Se  $X$ ,  $Y$  e  $Z$  formam uma cadeia de Markov nesta ordem ( $X \rightarrow Y \rightarrow Z$ ), então também formam uma cadeia de Markov na ordem inversa ( $Z \rightarrow Y \rightarrow X$ ).*

*Demonstração.*

$$p(x, y, z) = p(x)p(y|x)p(z|y) = p(x, y)p(z|y) \quad (98a)$$

$$= \frac{p(x, y)p(z|y)p(y)}{p(y)} = p(x|y)p(y, z) \quad (98b)$$

$$= p(x|y)p(y|z)p(z) \quad (98c)$$

□

A Desigualdade do Processamento de Dados estabelece que a informação mútua entre variáveis aleatórias não aumenta sob o processamento de dados. Em outras palavras, a informação relevante para uma variável não pode ser aumentada ao processar outra variável relacionada em uma cadeia de Markov.

**Teorema 12** (Desigualdade do Processamento de Dados) *Se  $X \rightarrow Y \rightarrow Z$  então*

$$I(X; Y) \geq I(X; Z) \quad (99)$$

*Demonstração.* Utilizando a regra da cadeia da informação mútua (Proposição 6), teremos

$$I(X; Y, Z) = I(X; Z) + I(X; Y|Z) \quad (100a)$$

$$= I(X; Y) + I(X; Z|Y) \quad (100b)$$

Como  $X \perp\!\!\!\perp Z|Y$  ( $X$  e  $Z$  são condicionalmente independentes, dado  $Y$ ), temos que  $I(X; Z|Y) = 0$ . Como  $I(X; Y|Z) \geq 0$ , usando eqs. (100a) e (100b), teremos

$$I(X; Z) + \underbrace{I(X; Y|Z)}_{\geq 0} = I(X; Y) + \overbrace{I(X; Z|Y)} \rightarrow 0 \quad (101)$$

Podemos concluir então que

$$I(X; Z) \leq I(X; Y). \quad (102)$$

□

Note que teremos igualdade na Equação (99) se, e somente se,  $I(X; Y|Z) = 0$  (i.e.  $X \rightarrow Z \rightarrow Y$ ). De forma similar, também podemos mostrar que  $I(Y; Z) \geq I(X; Z)$ .

**Corolário 12.1** *Se  $Z = g(Y)$ , então  $I(X; Y) \geq I(X; g(Y))$ .*

*Demonstração.*  $X \rightarrow Y \rightarrow g(Y)$  forma uma cadeia de Markov.

□

**Corolário 12.2** *Se  $X \rightarrow Y \rightarrow Z$  então  $I(X; Y|Z) \leq I(X; Y)$ .*

*Demonstração.*

$$\underbrace{I(X; Z) + I(X; Y|Z)}_{\geq 0} = I(X; Y) + \overbrace{I(X; Z|Y)} \rightarrow 0 \quad (103)$$

então

$$I(X; Y|Z) \leq I(X; Y). \quad (104)$$

□

Note também que

$$I(X; Y|Z) = H(X|Z) - H(X|Y, Z) \quad (105a)$$

$$\leq H(X) - H(X|Y) \quad (105b)$$

$$= I(X; Y). \quad (105c)$$

O processamento digital de imagem para remoção de ruído, por exemplo, é uma aplicação prática para a desigualdade de processamento de dados. Considere  $X$  como a imagem original,  $Y$  como a imagem ruidosa recebida em uma comunicação, e  $Z$  a imagem após o processamento para remoção de ruído. Podemos considerar que temos aqui uma cadeia de Markov  $X \rightarrow Y \rightarrow Z$  e assim podemos aplicar a desigualdade de processamento de dados, que nos diz:  $I(X; Z) \leq I(X; Y)$ . O processo de remoção de ruído não pode aumentar a informação mútua entre a imagem original  $X$  e a imagem processada  $Z$  além do que já existe entre  $X$  e  $Y$ , indicando uma perda inevitável de informação devido ao ruído. No entanto, a remoção de ruído é possível porque o algoritmo pode explorar regularidades estatísticas ou estruturais na imagem.

### Algumas Relações Notáveis

**Proposição 11** (Determinismo e a incerteza nula)  $H(X) = 0$  se, e somente se,  $X$  for determinístico.

*Demonstração.* Se  $X$  é determinístico, então existe  $x^* \in \mathcal{X}$  tal que  $p(x^*) = 1$  e  $p(x) = 0 \forall x \neq x^*$ . Neste caso, teremos  $H(X) = -p(x^*) \log p(x^*) = 0$ . Por outro lado, se  $X$  não for determinístico, então existe  $x^* \in \mathcal{X}$  tal que  $0 < p(x^*) < 1$ . Desta forma,  $H(X) \geq -p(x^*) \log p(x^*) > 0$ . Podemos concluir assim que  $H(X) = 0$  se e somente se  $X$  for determinístico.  $\square$

**Proposição 12** (Quando  $Y$  é uma função de  $X$ )  $H(Y|X) = 0$  se, e somente se,  $Y$  for uma função de  $X$ , ou seja,  $Y = g(X)$ .

*Demonstração.* Observando a Equação (33a), podemos concluir que  $H(Y|X) = 0$  se e somente se todos os termos do somatório forem nulos, ou seja,  $H(Y|X = x) = 0$  para cada  $x \in S_X$ . A partir da Proposição 11, sabemos que isto ocorre se, e somente se,  $Y$  é determinístico para cada  $x$  dado. Em outras palavras,  $Y$  é uma função de  $X$ .  $\square$

*Demonstração.* Uma outra forma de demonstração é por contradição. Assuma que existe  $x$ , digamos  $x_0$ , e dois valores diferentes de  $y$ , digamos  $y_1$  e  $y_2$ , tal que  $p(x_0, y_1) > 0$  e  $p(x_0, y_2) > 0$ . Então a marginal é  $p(x_0) \geq p(x_0, y_1) + p(x_0, y_2) > 0$ . Temos também

$$p(y_1|x_0) = \frac{p(x_0, y_1)}{p(x_0)} \text{ e } p(y_2|x_0) = \frac{p(x_0, y_2)}{p(x_0)} \quad (106)$$

então ambos  $p(y_1|x_0)$  e  $p(y_2|x_0)$  não são iguais a 0 (zero) ou 1 (um).

$$H(Y|X) = E[H(Y|X)] \quad (107a)$$

$$= - \sum_x p(x) H(Y|X = x) \quad (107b)$$

$$= - \sum_x p(x) \sum_y p(y|x) \log p(y|x) \quad (107c)$$

$$\geq -p(x_0) \sum_y p(y|x_0) \log p(y|x_0) \quad (107d)$$

$$\geq - \underbrace{p(x_0)}_{>0} \underbrace{[p(y_1|x_0) \log p(y_1|x_0) + p(y_2|x_0) \log p(y_2|x_0)]}_{<0} \quad (107e)$$

$$> 0 \quad (107f)$$

Então, a entropia condicional  $H(Y|X)$  é nula se e somente se  $Y$  for uma função de  $X$ . Se  $Y$  for uma função de  $X$ , teremos  $p(y_i|x_0) = 0$  ou 1, ou seja, a probabilidade  $p(y_i|x_0)$  será igual a 1 apenas para um  $y_i$  e zero para os demais.  $\square$

**Teorema 13** (Informação mútua condicional é não nula) *Para variáveis aleatórias  $X$ ,  $Y$  e  $Z$ ,*

$$I(X; Y|Z) \geq 0, \quad (108)$$

*com igualdade se, e somente se,  $X$  e  $Y$  forem independentes quando condicionados a  $Z$ , ou seja,  $X \perp\!\!\!\perp Y|Z$ .*

*Demonstração.* Observe que

$$I(X; Y|Z) = \sum_{x,y,z} p(x, y, z) \log \frac{p(x, y|z)}{p(x|z)p(y|z)} \quad (109a)$$

$$= \sum_z p(z) \sum_{x,y} p(x, y|z) \log \frac{p(x, y|z)}{p(x|z)p(y|z)} \quad (109b)$$

$$= \sum_z p(z) D(p_{XY|z} || p_{X|z} p_{Y|z}), \quad (109c)$$

onde  $p_{XY|z}$  representa  $\{p(x, y|z), (x, y) \in \mathcal{X} \times \mathcal{Y}\}$ ,  $p_{X|z}$  representa  $\{p(x|z), x \in \mathcal{X}\}$ , e  $p_{Y|z}$  representa  $\{p(y|z), y \in \mathcal{Y}\}$ . Dado  $z$ , ambos  $p_{XY|z}$  e  $p_{X|z}p_{Y|z}$  são distribuições conjuntas em  $\mathcal{X} \times \mathcal{Y}$ . Teremos então

$$D(p_{XY|z} || p_{X|z} p_{Y|z}) \geq 0. \quad (110)$$

Concluimos assim que  $I(X; Y|Z) \geq 0$ . A partir da Definição 5, observamos que  $I(X; Y|Z) = 0$  se, e somente se, para todo  $z \in S_z$ ,

$$p(x, y|z) = p(x|z)p(y|z) \quad (111a)$$

ou

$$p(x, y, z) = p(x, z)p(y|z) \quad (111b)$$

para todo  $x$  e  $y$ . Então  $X$  e  $Y$  são independentes quando condicionados a  $Z$ . □

**Proposição 13** (Independência e informação mútua nula)  $I(X; Y) = 0$  se, e somente se,  $X \perp\!\!\!\perp Y$ .

*Demonstração.* Este caso equivale ao Teorema 13 com a variável  $Z$  degenerada. □

**Teorema 14** (Condicionar não aumenta a entropia)

$$H(Y|X) \leq H(Y) \quad (112)$$

*com igualdade se, e somente se,  $X \perp\!\!\!\perp Y$ .*

*Demonstração.* Basta considerar

$$H(Y|X) = H(Y) - \underbrace{I(X; Y)}_{\geq 0} \leq H(Y). \quad (113)$$

A igualdade ocorrerá se, e somente se,  $I(X; Y) = 0$ , o que equivale a  $X \perp\!\!\!\perp Y$ , conforme a Proposição 13. □

A próxima desigualdade estabelece um teto para a entropia conjunta de um conjunto de variáveis aleatórias, assumindo que elas são independentes.

**Teorema 15** (Limite de Independência para Entropia) *A entropia conjunta  $H(X_1, X_2, \dots, X_n)$  de um conjunto de variáveis aleatórias  $X_1, X_2, \dots, X_n$  é sempre menor ou igual à soma das entropias individuais, ou seja,*

$$H(X_1, X_2, \dots, X_n) \leq \sum_{i=1}^n H(X_i), \quad (114)$$

*com igualdade se, e somente se,  $X_i$ ,  $i = 1, 2, \dots, n$ , forem mutuamente independentes.*

*Demonstração.*

$$H(X_1, X_2, \dots, X_n) = \sum_{i=1}^n H(X_i | X_1, \dots, X_{i-1}) \quad (115a)$$

$$\leq \sum_{i=1}^n H(X_i), \quad (115b)$$

onde em 115a utilizamos a regra da cadeia e em 115b utilizamos que condicionar não aumenta a entropia. A desigualdade é justa se e somente se for justa para cada  $i$  em 115b, ou seja,

$$H(X_i | X_1, \dots, X_{i-1}) = H(X_i), \quad (116)$$

para  $i = 1, \dots, n$ . Isto equivale a ter  $X_i \perp X_j$ ,  $\forall i \neq j$ .

□

**Teorema 16**

$$I(X; Y, Z) \geq I(X; Y), \quad (117)$$

*com igualdade se, e somente se, tivermos uma cadeia de Markov  $X \rightarrow Y \rightarrow Z$ .*

*Demonstração.* Pela regra da cadeia da informação mútua obtemos

$$I(X; Y, Z) = I(X; Y) + I(X; Z | Y) \geq I(X; Y), \quad (118)$$

onde, para obter a desigualdade, consideramos o fato de que qualquer informação mútua é não negativa. Teremos  $I(X; Z | Y) = 0$  se  $X \rightarrow Y \rightarrow Z$  formar uma cadeia de Markov.

□

Note que o Teorema 16 não contradiz o Corolário 12.2. Quando existir uma cadeia de Markov  $X \rightarrow Y \rightarrow Z$ , teremos simplesmente  $I(X; Y, Z) = I(X; Y)$ , mostrando que  $Z$  não adiciona informação sobre  $X$  além do que  $Y$  já fornece, uma propriedade que reflete a estrutura de dependência da cadeia de Markov.

Veremos agora uma relação importante que conecta a entropia à probabilidade de colisão entre amostras independentes, sendo útil em aplicações como compressão de dados e análise de colisões em funções hash, além de reforçar a propriedade da distribuição uniforme como maximizadora da entropia.

**Proposição 14** (Limite inferior da probabilidade de colisão) *Considere duas variáveis aleatórias i.i.d.  $X$  e  $X'$  com entropia  $H(X)$ . Teremos o seguinte limite inferior para a probabilidade de colisão ( $X = X'$ ):*

$$Pr(X = X') \geq 2^{-H(X)} \quad (119)$$

com igualdade se e somente se  $X$  possuir distribuição uniforme.

*Demonstração.* Podemos calcular explicitamente o valor probabilidade de coincidência para duas variáveis aleatórias:

$$Pr(X = X') = Pr(X = x_1 | X' = x_1)Pr(X' = x_1) + \dots + Pr(X = x_n | X' = x_n)Pr(X' = x_n) \quad (120a)$$

$$= Pr(X = x_1)Pr(X' = x_1) + \dots + Pr(X = x_n)Pr(X' = x_n) \quad (120b)$$

$$= p^2(x_1) + \dots + p^2(x_n) = \sum_x p^2(x). \quad (120c)$$

Em muitas aplicações práticas, a distribuição exata  $p$  pode não ser conhecida, mas a entropia pode ser estimada. Torna-se também intuitivo buscar interpretar a probabilidade de coincidência em termos de incerteza. Para tanto, suponha que  $X \sim p(x)$ . Usando a desigualdade de Jensen temos

$$2^{E[\log p(X)]} \leq E[2^{\log p(X)}], \quad (121)$$

pois  $2^x$  é convexa. Logo,

$$2^{-H(X)} = 2^{\sum_x p(x) \log p(x)} = 2^{E[\log p(X)]} \quad (122a)$$

$$\leq E[2^{\log p(X)}] \quad (122b)$$

$$= \sum_x p(x) 2^{\log p(x)} = \sum_x p(x)p(x) \quad (122c)$$

$$= \sum_x p^2(x) = Pr(X = X') \quad (122d)$$

□

Note que, para maximizar a probabilidade  $Pr(X = X')$ , devemos minimizar a entropia. No limite, quando  $H(X) = 0$ , teremos  $Pr(X = X') \geq 1$ , logo será igual a 1 e assim  $X = X'$  sem dúvida. Por outro lado, maximizar a entropia garante que este limite inferior será mínimo. Isso apenas nos garante que há margem para minimizar

a probabilidade de colisão até determinado ponto, não restringe superiormente esta probabilidade, o que seria desejável de se encontrar.

A Equação (120c) fornece a probabilidade de colisão.

**Definição 9** (Entropia de Colisão). A chamada entropia de colisão, ou entropia de Rényi de segunda ordem ( $n = 2$ ) é definida com

$$H_2(X) = -\log(\Pr(X = X')). \quad (123)$$

$H_2(X)$  mede a incerteza de que duas realizações de dois experimentos aleatórios independentes  $X$  e  $X'$  tenham o mesmo resultado. Muitas das propriedades básicas da entropia de Shannon não podem ser aplicadas à entropia de Rényi. Para mais informações sobre este tópico, recomendamos a leitura de Delfs e Knebl (2015).

Vamos analisar agora a probabilidade de colisão quando as duas variáveis aleatórias possuem distribuições distintas.

**Corolário 16.1** (Limite inferior da probabilidade de colisão com distribuições distintas) *Seja  $X, X'$  independentes com  $X \sim p(x)$  e  $X' \sim q(x)$ ,  $x, x' \in \mathcal{X}$ , então*

$$\Pr(X = X') \geq 2^{-H(p) - D(p||q)} \quad (124a)$$

$$\Pr(X = X') \geq 2^{-H(q) - D(q||p)} \quad (124b)$$

ou seja,

$$\Pr(X = X') \geq \max\left(2^{-H(p) - D(p||q)}, 2^{-H(q) - D(q||p)}\right) \quad (125)$$

*Demonstração.*

$$2^{-H(p) - D(p||q)} = 2^{\sum_x p(x) \log p(x) + \sum_x p(x) \log \frac{q(x)}{p(x)}} \quad (126a)$$

$$= 2^{\sum_x p(x) \log q(x)} \quad (126b)$$

$$= 2^{E_p[\log q(X)]} \quad (126c)$$

$$\leq \sum_x p(x) 2^{\log q(x)} \quad (126d)$$

$$= \sum_x p(x) q(x) \quad (126e)$$

$$= \Pr(X = X'), \quad (126f)$$

onde em 126d aplicamos a desigualdade de Jensen. □

### Comunicação em um canal ruidoso

A comunicação em um canal ruidoso é um cenário clássico em teoria da informação, onde uma mensagem  $X$  é transmitida através de um canal sujeito a ruído, resultando em uma versão distorcida  $Y$ . O receptor, tenta estimar  $X$  a partir de  $Y$ , gerando uma estimativa  $\hat{X} = g(Y)$ .

Esse processo forma uma cadeia de Markov:  $X \rightarrow Y \rightarrow \hat{X}$ . O processo de estimação está inevitavelmente sujeito a erros. Para quantificar esse erro e estabelecer limites sobre a probabilidade de erro na estimação de  $X$ , introduziremos a desigualdade de Fano.

Primeiramente, definiremos o erro na estimação como o evento em que  $X \neq \hat{X}$ , e a probabilidade de erro será denotada por  $P_e = p(X \neq \hat{X})$ . Em muitas situações, não é possível calcular diretamente  $P_e$ ; no entanto, veremos que a incerteza sobre  $X$  dado  $Y$ , representada por  $H(X|Y)$ , pode ser usada para estabelecer um limite inferior para essa probabilidade de erro. Para derivar esse limite, será suficiente conhecer as características do canal, ou seja, a distribuição condicional  $p(y|x)$ , que descreve a relação entre a entrada  $X$  e a saída  $Y$ .

**Teorema 17** (Desigualdade de Fano) *Para qualquer estimador  $\hat{X}$  tal que  $X \rightarrow Y \rightarrow \hat{X}$ , com  $P_e = \Pr(X \neq \hat{X})$ , temos*

$$H(P_e) + P_e \log(|\mathcal{X}| - 1) \geq H(X|\hat{X}) \geq H(X|Y). \quad (127)$$

Esta desigualdade pode ser simplificada<sup>10</sup> na forma

$$P_e \geq \frac{H(X|Y) - 1}{\log(|\mathcal{X}| - 1)}, \quad (129)$$

onde utilizamos  $H(P_e) \leq 1$ .

*Demonstração.* Vamos primeiramente definir uma função de erro:

$$E = \begin{cases} 1 & , \text{ se } \hat{X} \neq X (\text{erro}) \\ 0 & , \text{ se } \hat{X} = X (\text{sem erro}). \end{cases} \quad (130)$$

Utilizando a regra da cadeia obtemos:

$$H(E, X|\hat{X}) = H(X|\hat{X}) + \underbrace{H(E|X, \hat{X})}_{=0} \quad (131a)$$

$$= \underbrace{H(E|\hat{X})}_{\leq H(E)=H(P_e)} + \underbrace{H(X|E, \hat{X})}_{\leq P_e \log(|\mathcal{X}| - 1)}. \quad (131b)$$

O termo  $H(E|X, \hat{X})$  é nulo pois o erro é uma função determinística de  $X$  e  $\hat{X}$ , então, sabendo  $X$  e  $\hat{X}$ , determinamos  $E$ . Como condicionar não aumenta a entropia, usamos  $H(E|\hat{X}) \leq H(E) = H(P_e)$ . E por fim, note que

$$H(X|\hat{X}, E) = p(E=0) \underbrace{H(X|\hat{X}, E=0)}_{=0} + p(E=1)H(X|\hat{X}, E=1) \quad (132a)$$

$$= (1 - P_e)0 + P_e H(X|\hat{X}, E=1) \leq P_e \log(|\mathcal{X}| - 1), \quad (132b)$$

<sup>10</sup> Para o caso de um alfabeto binário ( $|\mathcal{X}| = 2$ ), a desigualdade de Fano na forma da Equação (129) não poderá ser aplicada. Devemos então utilizar a forma mais relaxada:

$$P_e \geq \frac{H(X|Y) - 1}{\log(|\mathcal{X}| - 1)} > \frac{H(X|Y) - 1}{\log |\mathcal{X}|} \quad (128)$$



onde usamos que, se não há erro ( $E = 0$ ) e conhecemos  $\hat{X}$ , então determinamos  $X$ . Não existe entropia residual em  $X$  quando é dado  $\hat{X}$  e  $E = 0$ . Logo  $H(X|\hat{X}, E = 0) = 0$ . Usamos ainda que se conhecemos  $\hat{X}$  e existe um erro ( $E = 1$ ), então sabemos que  $X$  é diferente de  $\hat{X}$ , logo isto nos deixa com  $(|\mathcal{X}| - 1)$  alternativas. Assim, teremos  $H(X|\hat{X}, E = 1) \leq \log(|\mathcal{X}| - 1)$ . Aplicando  $H(X|\hat{X}, E) \leq P_e \log(|\mathcal{X}| - 1)$  na Equação (131), obtemos agora

$$H(X|\hat{X}) = H(E|\hat{X}) + H(X|E, \hat{X}) \leq H(P_e) + P_e \log(|\mathcal{X}| - 1). \quad (133a)$$

Como  $X \rightarrow Y \rightarrow \hat{X}$  é uma cadeia de Markov, podemos utilizar a desigualdade de processamento de dados:

$$I(X; Y) \geq I(X; \hat{X}) \quad (134a)$$

$$H(X) - H(X|Y) \geq H(X) - H(X|\hat{X}) \quad (134b)$$

$$H(X|\hat{X}) \geq H(X|Y) \quad (134c)$$

Então, utilizando as Equações 133 e 134, obtemos

$$H(P_e) + P_e \log(|\mathcal{X}| - 1) \geq H(X|\hat{X}) \geq H(X|Y). \quad (135)$$

□

**Exemplo 4.** Considere uma variável aleatória discreta  $X \in \mathcal{X} = \{1, 2, \dots, 5\}$  com função massa de probabilidade  $p(x) = (0.35, 0.35, 0.1, 0.1, 0.1)$ . Seja  $Y \in \mathcal{Y} = \{1, 2\}$ , de forma que, se  $x \leq 2$  teremos  $y = x$  com probabilidade  $6/7$  e, se  $x > 2$ , teremos  $y = 1$  ou  $2$  com igual probabilidade. A melhor estratégia é utilizar o estimador  $\hat{x} = y$ . Calcule a probabilidade de erro e o limite dado pela desigualdade de Fano.

A distribuição condicional  $p(y|x)$  é

X \ Y	Y	
	1	2
1	6/7	1/7
2	1/7	6/7
3	1/2	1/2
4	1/2	1/2
5	1/2	1/2

A efetiva probabilidade de erro é dada por

$$P_e = 1 - P_a \quad (136a)$$

$$= 1 - \sum_{i=1}^5 P(x_i = y_i) \quad (136b)$$

$$= 1 - (p(y = 1|x = 1)p(x = 1) + p(y = 2|x = 2)p(x = 2) + 0 + 0 + 0) \quad (136c)$$

$$= 1 - \left( \frac{6}{7} 0.35 + \frac{6}{7} 0.35 \right) = 0.4 = \frac{2}{5} \quad (136d)$$

onde denotamos por  $P_a$  a probabilidade de acerto.

A desigualdade de Fano fornece um limite inferior pra a probabilidade de erro (predição incorreta do valor de  $X$  baseado em  $Y$ ). Este limite inferior é determinado pela incerteza remanescente  $H(X|Y)$  sobre  $X$  quando  $Y$  é conhecido.

Pelo teorema da desigualdade de Fano temos que

$$P_e \geq \frac{H(X|Y) - 1}{\log(|\mathcal{X}| - 1)} \quad (137)$$

$$H(X|Y) = - \sum_{x,y} p(x,y) \log p(x|y) \quad (138a)$$

$$= - \sum_{x,y} p(y|x)p(x) \log p(x|y) \quad (138b)$$

onde  $p(y|x)$  e  $p(x)$  são dados do problema e ainda será necessário calcular  $p(x|y)$  para encontrar  $H(X|Y)$ .

$$P(X|Y=1) = \frac{P(X, Y=1)}{P(Y=1)} = \frac{P(Y=1|X)P(X)}{P(Y=1)} \quad (139a)$$

$$= \frac{(\frac{6}{7}, \frac{1}{7}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}) \cdot (0.35, 0.35, 0.1, 0.1, 0.1)}{\sum ((\frac{6}{7}, \frac{1}{7}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}) \cdot (0.35, 0.35, 0.1, 0.1, 0.1))} \quad (139b)$$

$$= \frac{(0.3, 0.05, 0.05, 0.05, 0.05)}{1/2} \quad (139c)$$

$$= (0.6, 0.1, 0.1, 0.1, 0.1) \quad (139d)$$

$$P(X|Y=2) = \frac{(\frac{1}{7}, \frac{6}{7}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}) \cdot (0.35, 0.35, 0.1, 0.1, 0.1)}{\sum ((\frac{1}{7}, \frac{6}{7}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}) \cdot (0.35, 0.35, 0.1, 0.1, 0.1))} \quad (140a)$$

$$= (0.1, 0.6, 0.1, 0.1, 0.1) \quad (140b)$$

Desta forma Teremos

$$H(X|Y) = H(X|Y=1)P(Y=1) + H(X|Y=2)P(Y=2) \quad (141a)$$

$$= -\frac{1}{2} (0.6 \log 0.6 + 4 \times 0.1 \log 0.1) - \frac{1}{2} (4 \times 0.1 \log 0.1 + 0.6 \log 0.6) \quad (141b)$$

$$= 1.771 \text{ bits.} \quad (141c)$$

Utilizando a desigualdade de Fano

$$P_e \geq \frac{H(X|Y) - 1}{\log(|\mathcal{X}| - 1)} = \frac{1.771 - 1}{\log(5 - 1)} = 0.3855. \quad (142)$$

De fato, temos  $P_e = 0.4 \geq 0.3855$ .

## Medida de Informação

Exploramos conceitos como entropia, informação mútua e divergência de Kullback-Leibler sob a perspectiva de variáveis aleatórias e suas distribuições, mas essas grandezas podem ser entendidas de forma mais profunda através de um paralelo com a teoria dos conjuntos e a teoria da medida. Desta forma, buscaremos uma interpretação geométrica e estrutural para relações como a regra da cadeia e a desigualdade do processamento de dados, permitindo uma compreensão mais ampla de como a informação é estruturada e quantificada em espaços probabilísticos.

Dado um conjunto de variáveis aleatórias:  $X_1, X_2, \dots, X_n$ , a cada uma delas associamos um conjunto  $\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_n$ .

**Definição 10** (Campo). Um campo  $\mathcal{F}_n$  gerado pelos conjuntos  $\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_n$  é a coleção de conjuntos que podem ser obtidos através de qualquer sequência de operações usuais de conjuntos (união, interseção, complemento, e diferença) sobre os conjuntos  $\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_n$ .

**Definição 11** (Átomo). Os átomos de  $\mathcal{F}_n$  são os conjuntos da forma  $\cap_{i=1}^n Y_i$ , onde  $Y_i$  é  $\tilde{X}_i$  ou  $\tilde{X}_i^c$ , o complemento de  $\tilde{X}_i$ .

**Exemplo 5.** Vamos considerar o seguinte exemplo com  $n = 2$ . Neste caso, teremos os conjuntos  $\tilde{X}_1, \tilde{X}_2$  e seus complementos, respectivamente,  $\tilde{X}_1^c, \tilde{X}_2^c$ . Existirão 4 átomos:  $\tilde{X}_1 \cap \tilde{X}_2$ ,  $\tilde{X}_1 \cap \tilde{X}_2^c$ ,  $\tilde{X}_1^c \cap \tilde{X}_2$ , e  $\tilde{X}_1^c \cap \tilde{X}_2^c$ , que estão representados na Figura 11.

**Exemplo 6.** Considerando agora o caso com  $n = 3$ . Teremos os conjuntos  $\tilde{X}_1, \tilde{X}_2, \tilde{X}_3$  e seus complementos, respectivamente,  $\tilde{X}_1^c, \tilde{X}_2^c, \tilde{X}_3^c$ . Existirão 8 átomos:  $\tilde{X}_1 \cap \tilde{X}_2 \cap \tilde{X}_3$ ,  $\tilde{X}_1 \cap \tilde{X}_2 \cap \tilde{X}_3^c$ ,  $\tilde{X}_1 \cap \tilde{X}_2^c \cap \tilde{X}_3$ ,  $\tilde{X}_1 \cap \tilde{X}_2^c \cap \tilde{X}_3^c$ ,  $\tilde{X}_1^c \cap \tilde{X}_2 \cap \tilde{X}_3$ ,  $\tilde{X}_1^c \cap \tilde{X}_2 \cap \tilde{X}_3^c$ ,  $\tilde{X}_1^c \cap \tilde{X}_2^c \cap \tilde{X}_3$ , e  $\tilde{X}_1^c \cap \tilde{X}_2^c \cap \tilde{X}_3^c$ , como representados na Figura 12.

Consideremos  $n$  variáveis aleatórias discretas  $X_1, X_2, \dots, X_n$ , onde cada  $X_i$  tem um suporte finito  $S_{X_i}$ , com  $|S_{X_i}| = k_i$ , e os suportes  $S_{X_i}$  podem ser distintos. Por exemplo,  $S_{X_1} = \{0, 1\}$ ,  $S_{X_2} = \{a, b, c\}$ , e assim por diante. O espaço amostral é o produto cartesiano  $\Omega = S_{X_1} \times S_{X_2} \times \dots \times S_{X_n}$ , com  $|\Omega| = k_1 \cdot k_2 \cdot \dots \cdot k_n$ , e o campo  $\mathcal{F}_n = 2^\Omega$  contém todos os subconjuntos de  $\Omega$ . Nesse contexto, os átomos são os eventos  $\{(X_1, \dots, X_n) = (x_1, \dots, x_n)\}$ , e grandezas como a entropia  $H(X_1, \dots, X_n)$  podem ser interpretadas como medidas sobre  $\mathcal{F}_n$ . Para ilustrar propriedades específicas, consideremos o caso particular em que  $S_{X_i} = \{0, 1\}, \forall i$ , ou seja,  $\Omega = \{0, 1\}^n$ , com  $|\Omega| = 2^n$  e  $\mathcal{F}_n = 2^\Omega$ . Nesse cenário, teremos as seguintes propriedades:

1. há exatamente  $2^n$  átomos;
2.  $|\mathcal{F}_n| = 2^\Omega = 2^{2^n}$ ;

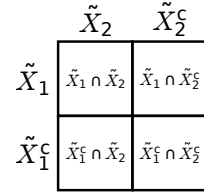


Figura 11: Exemplo de átomos para  $n = 2$ .

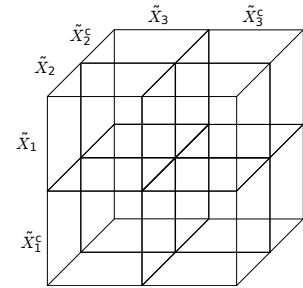


Figura 12: Exemplo de átomos para  $n = 3$ .

3. Todos os átomos em  $F_n$  são disjuntos;
4. Todo conjunto  $A \in F_n$  pode ser expresso de forma única como uma união de um subconjunto dos átomos.

Em análise matemática, uma medida em um conjunto  $S$  é uma forma sistemática de atribuir números a todo subconjunto de  $S$ , sendo intuitivamente interpretada como o seu tamanho. Medida com sinal é uma generalização do conceito de medida permitindo que esta assuma valores negativos.

**Definição 12** (Medida com sinal). Uma função real  $\mu$  definida em  $\mathcal{F}_n$  é chamada medida com sinal se for aditiva no conjunto, i.e., para  $A$  e  $B$  disjuntos em  $\mathcal{F}_n$ ,

$$\mu(A \cup B) = \mu(A) + \mu(B). \quad (143)$$

Para uma medida com sinal  $\mu$  teremos  $\mu(\emptyset) = 0$ , já que  $\mu(A) = \mu(A \cup \emptyset) = \mu(A) + \mu(\emptyset)$ .

Uma medida com sinal  $\mu$  em  $\mathcal{F}_n$  é completamente especificada por seus valores nos átomos de  $\mathcal{F}_n$ . Os valores de  $\mu$  em outros conjuntos de  $\mathcal{F}_n$  podem ser obtidos pela aditividade de conjuntos, já que qualquer  $\tilde{X} \in \mathcal{F}_n$  pode ser representado como  $\tilde{X} = \cup_{i=1}^n Y_i$ , onde  $Y_i$  são átomos escolhidos apropriadamente.

**Exemplo 7** ( $n = 2$ ). Uma medida com sinal  $\mu$  em  $\mathcal{F}_2$  é completamente especificada pelos valores  $\mu(\tilde{X}_1 \cap \tilde{X}_2)$ ,  $\mu(\tilde{X}_1 \cap \tilde{X}_2^c)$ ,  $\mu(\tilde{X}_1^c \cap \tilde{X}_2)$ , e  $\mu(\tilde{X}_1^c \cap \tilde{X}_2^c)$ .

O valor de  $\mu$  em  $\tilde{X}_1$  pode ser obtido da seguinte forma

$$\mu(\tilde{X}_1) = \mu((\tilde{X}_1 \cap \tilde{X}_2) \cup (\tilde{X}_1 \cap \tilde{X}_2^c)) \quad (144a)$$

$$= \mu(\tilde{X}_1 \cap \tilde{X}_2) + \mu(\tilde{X}_1 \cap \tilde{X}_2^c). \quad (144b)$$

O valor de  $\mu$  em  $\tilde{X}_1 \setminus \tilde{X}_2$  é dado por

$$\mu(\tilde{X}_1 \setminus \tilde{X}_2) = \mu(\tilde{X}_1 \cap \tilde{X}_2^c). \quad (145)$$

O valor de  $\mu$  em  $\tilde{X}_1 \cup \tilde{X}_2$  pode ser obtido através de

$$\mu(\tilde{X}_1 \cup \tilde{X}_2) = \mu((\tilde{X}_1 \cap \tilde{X}_2) \cup (\tilde{X}_1 \cap \tilde{X}_2^c) \cup (\tilde{X}_1^c \cap \tilde{X}_2)) \quad (146a)$$

$$= \mu(\tilde{X}_1 \cap \tilde{X}_2) + \mu(\tilde{X}_1 \cap \tilde{X}_2^c) + \mu(\tilde{X}_1^c \cap \tilde{X}_2) \quad (146b)$$

Os conjuntos  $\tilde{X}_1$  e  $\tilde{X}_2$  estão associados às variáveis aleatórias  $X_1$  e  $X_2$ . O campo  $\mathcal{F}_2$  é gerado por  $\tilde{X}_1$  e  $\tilde{X}_2$ , através dos átomos  $(\tilde{X}_1 \cap \tilde{X}_2)$ ,  $(\tilde{X}_1 \cap \tilde{X}_2^c)$ ,  $(\tilde{X}_1^c \cap \tilde{X}_2)$ , e  $(\tilde{X}_1^c \cap \tilde{X}_2^c)$ . O diagrama de informação é apresentado na Figura 13.

O conjunto universo será considerada como sendo  $\Omega = \tilde{X}_1 \cup \tilde{X}_2$ . Desta forma, o átomo  $\tilde{X}_1^c \cap \tilde{X}_2^c$  se degenera ao conjunto vazio,

$$\tilde{X}_1^c \cap \tilde{X}_2^c = (\tilde{X}_1 \cup \tilde{X}_2)^c = \Omega^c = \emptyset. \quad (147)$$

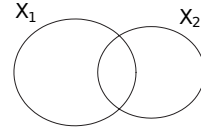


Figura 13: Diagrama de informação para  $X_1$  e  $X_2$ .

Para as v.a.s  $X_1$  e  $X_2$ , as medidas de informação de Shannon são

$$H(X_1), H(X_2), H(X_1|X_2), H(X_2|X_1), H(X_1, X_2), I(X_1; X_2). \quad (148)$$

Utilizando a notação  $A \cap B^c \equiv A \setminus B$ , definimos uma medida com sinal  $\mu^*$

$$\mu^*(\tilde{X}_1 \setminus \tilde{X}_2) = H(X_1|X_2), \quad (149a)$$

$$\mu^*(\tilde{X}_2 \setminus \tilde{X}_1) = H(X_2|X_1), \quad (149b)$$

$$\mu^*(\tilde{X}_1 \cap \tilde{X}_2) = I(X_1; X_2). \quad (149c)$$

Estes são os valores de  $\mu^*$  nos átomos não vazios de  $\mathcal{F}_2$ . Os valores de  $\mu^*$  nos demais conjuntos de  $\mathcal{F}_2$  podem ser obtidos por adição de conjuntos. Em particular, temos as relações

$$\mu^*(\tilde{X}_1 \cup \tilde{X}_2) = H(X_1, X_2), \quad (150a)$$

$$\mu^*(\tilde{X}_1) = H(X_1), \quad (150b)$$

$$\mu^*(\tilde{X}_2) = H(X_2). \quad (150c)$$

Por exemplo, a Equação 150a pode ser verificada

$$\begin{aligned} \mu^*(\tilde{X}_1 \cup \tilde{X}_2) &= \mu^*((\tilde{X}_1 \setminus \tilde{X}_2) \cup (\tilde{X}_2 \setminus \tilde{X}_1) \cup (\tilde{X}_1 \cap \tilde{X}_2)) \\ &= \mu^*(\tilde{X}_1 \setminus \tilde{X}_2) + \mu^*(\tilde{X}_2 \setminus \tilde{X}_1) + \mu^*(\tilde{X}_1 \cap \tilde{X}_2) \\ &= H(X_1|X_2) + H(X_2|X_1) + I(X_1; X_2) \\ &= H(X_1, X_2). \end{aligned} \quad (151a)$$

A Equação 150b também pode ser facilmente verificada

$$\begin{aligned} \mu^*(\tilde{X}_1) &= \mu^*((\tilde{X}_1 \cap \tilde{X}_2) \cup (\tilde{X}_1 \cap \tilde{X}_2^c)) \\ &= \mu^*(\tilde{X}_1 \cap \tilde{X}_2) + \mu^*(\tilde{X}_1 \cap \tilde{X}_2^c) \\ &= I(X_1; X_2) + H(X_1|X_2) = H(X_1). \end{aligned} \quad (152a)$$

É possível então verificar a seguinte correspondência com as medidas de informação de Shannon

$$H/I \leftrightarrow \mu^* \quad (153a)$$

$$, \leftrightarrow \cup \quad (153b)$$

$$; \leftrightarrow \cap \quad (153c)$$

$$| \leftrightarrow \setminus \quad (153d)$$

Com a notação de medida, não existe distinção entre  $H$  e  $I$ , podemos escrever  $H(X; Y) = I(X; Y)$ , utilizando a notação do ponto-e-vírgula.

# Propriedade da Equipartição Assintótica

A Propriedade da Equipartição Assintótica (AEP) visa analisar o comportamento de sequências no limite, quando estas sequências tornam-se muito grandes. A AEP mostra que para uma sequência longa de variáveis aleatórias independentes e identicamente distribuídas (i.i.d.), a probabilidade de observar uma sequência típica é aproximadamente  $2^{-nH(X)}$ , onde  $H(X)$  é a entropia da fonte e  $n$  o comprimento da sequência.

Primeiramente, vamos rever o conceito de estatística de uma amostra.

**Definição 13** (Estatística amostral). Seja  $x_1, x_2, \dots, x_n$  uma sequência de comprimento  $n$  de valores observados de uma amostra de tamanho  $n$ , obtidos a partir da realização de variáveis aleatórias  $X_1, X_2, \dots, X_n$ , uma estatística de uma amostra é qualquer função calculada a partir de uma amostra de dados,  $T(x_1, x_2, \dots, x_n)$ .

Exemplos comuns incluem a média amostral

$$\bar{x} = T(x_1, \dots, x_n) = \frac{1}{n} \sum_{i=1}^n x_i, \quad (154)$$

a variância amostral

$$s^2 = T(x_1, \dots, x_n) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2, \quad (155)$$

a mediana amostral, a primeira amostra  $x_1$ , a última amostra  $x_n$ , ou até estatísticas como o máximo ou mínimo da amostra, que capturam aspectos específicos dos dados observados. Quando avaliamos  $T$  nas variáveis aleatórias  $X_1, X_2, \dots, X_n$ , o resultado  $T(X_1, X_2, \dots, X_n)$  é uma variável aleatória, com sua própria distribuição.

**Exemplo 8** (Ensaio de Bernoulli). Considere o experimento de Bernoulli com  $X_1, \dots, X_n$  i.i.d., com  $X_i \in \{0, 1\}$  e parâmetro  $\theta = Pr(X_i = 1)$ .

Uma dada sequência qualquer  $x_1, \dots, x_n$  terá então probabilidade dada por

$$p(x_1, x_2, \dots, x_n) = \prod_{i=1}^n \theta^{x_i} (1 - \theta)^{1-x_i} = \theta^{\sum_i x_i} (1 - \theta)^{n - \sum_i x_i} \quad (156)$$

Considere agora a seguinte estatística

$$T(x_1, \dots, x_n) = \sum_{i=1}^n x_i, \quad (157)$$

o somatório da amostra, no caso, a quantidade de valores iguais a 1 que apareceu em uma realização. Uma vez que sabemos tal estatísticas, a probabilidade de uma sequência pode ser expressa sem fazer referência à  $\theta$  (parâmetro que caracteriza a distribuição).

$$p(x_1, \dots, x_n | T(x_1, \dots, x_n), \theta) = p(x_1, \dots, x_n | T(x_1, \dots, x_n)) \quad (158a)$$

$$= \begin{cases} \frac{1}{\binom{n}{k}} & , \sum_i x_i = k \\ 0 & , \text{caso contrário.} \end{cases} \quad (158b)$$

Em outras palavras,  $X_{1:N} \perp\!\!\!\perp \theta | T(X_{1:N})$ . Isto implica na cadeia de Markov  $\theta \rightarrow T(X_{1:N}) \rightarrow X_{1:N}$ . Aplicando a desigualdade de processamento de dados, obtemos

$$I(\theta; T(X_{1:N})) \geq I(\theta; X_{1:N}). \quad (159)$$

Por outro lado, sabemos que  $T(X_{1:N})$  é uma função de  $X_{1:N}$ . Desta forma, também temos a seguinte cadeia de Markov:  $\theta \rightarrow X_{1:N} \rightarrow T(X_{1:N})$ . Novamente, aplicando a desigualdade de processamento de dados, obtemos

$$I(\theta; X_{1:N}) \geq I(\theta; T(X_{1:N})). \quad (160)$$

Então, para que 159 e 160 sejam satisfeitos, devemos ter

$$I(\theta; X_{1:N}) = I(\theta; T(X_{1:N})), \quad (161)$$

e nenhuma informação é perdida sobre  $\theta$  indo de  $X_{1:N}$  para  $T(X_{1:N})$ .

**Definição 14** (Estatística Suficiente). Uma função  $T(\cdot)$  é dita ser uma estatística suficiente em relação à família  $\{f_\theta(x)\}$  se  $X$  é independente de  $\theta$  dado  $T(X)$  para qualquer distribuição em  $\theta$  (i.e.  $\theta \rightarrow T(X) \rightarrow X$  forma uma cadeia de Markov). Então

$$I(\theta; X) = I(\theta; T(X)), \quad \forall \theta \quad (162)$$

Uma estatística suficiente preserva a informação mútua e reciprocamente

$$X \perp\!\!\!\perp \theta | T(X). \quad (163)$$

Podemos verificar que, no Exemplo 8, a estatísticas utilizada é uma estatísticas suficiente (veja a Equação (158)).

Um critério prático para identificar estatísticas suficientes é dado pelo Teorema da Fatoração de Fisher-Neyman.

**Teorema 18** *Uma estatísticas  $T(X_1, \dots, X_N)$  é suficiente para  $\theta$  se, e somente se, a função de verossimilhança da amostra pode ser escrita como:*

$$\mathcal{L}(\theta; x_1, \dots, x_n) = g(T(x_1, \dots, x_n), \theta) \cdot h(x_1, \dots, x_n), \quad (164)$$

onde  $g$  é uma função que depende de  $\theta$  e da estatística  $T$ , e  $h$  é uma função que não depende de  $\theta$ .

A demonstração e mais informações sobre o tema podem ser vistos em Berger e Casella (2002).

O histograma empírico da amostra é uma estatística que descreve a distribuição de frequências relativas dos valores observados em uma amostra.

**Definição 15** (Histograma Empírico). Para uma amostra  $x_1, x_2, \dots, x_n$ , obtida a partir de variáveis aleatórias  $X_1, X_2, \dots, X_n$ , com suporte finito  $S_X = \{a_1, a_2, \dots, a_D\}$ , o histograma empírico pode ser definido como a função que associa cada símbolo  $a \in S_X$  à sua frequência relativa:

$$\hat{p}_n(a) = \frac{1}{n} \sum_{i=1}^n I\{x_i = a\}, \quad (165)$$

onde  $I\{x_i = a\}$  é a função indicadora

$$I\{x_i = a\} = \begin{cases} 1, & x = a, \\ 0, & x \neq a. \end{cases} \quad (166)$$

Desta forma,  $\hat{p}_n(a)$  representa a proporção de vezes que o valor  $a$  aparece na amostra.

Equivalentemente, o histograma empírico pode ser representado como a enúpla que contém as frequências relativas de todos os símbolos de  $S_X$ .

$$P_{x_{1:n}} \triangleq \left( \frac{N(a_1|x_{1:n})}{n}, \frac{N(a_2|x_{1:n})}{n}, \dots, \frac{N(a_D|x_{1:n})}{n} \right), \quad (167)$$

onde  $N(a_i|x_{1:n})$  é a contagem do número de ocorrências do símbolo  $a_i$  na amostra  $x_{1:n}$ . O histograma é uma estatística, já que é uma função da amostra. Podemos mostrar ainda que o histograma empírico é uma estatísticas suficiente:

$$p(x_{1:n}|P_{x_{1:n}}, \theta) = \begin{cases} \frac{1}{\binom{n}{n_1, n_2, \dots, n_D}} & , n_i = nP_{x_{1:n}}(a_i), \forall i \\ 0 & , \text{caso contrário} \end{cases} \quad (168a)$$

$$= p(x_{1:n}|P_{x_{1:n}}) \quad (168b)$$

onde temos o coeficiente multinomial<sup>11</sup>  $\binom{n}{k_1, k_2, \dots, k_m} = \frac{n!}{k_1! k_2! \dots k_m!}$ . Podemos observar que  $p(x_{1:n}|P_{x_{1:n}}, \theta) = p(x_{1:n}|P_{x_{1:n}})$ , ou seja, é independente de  $\theta$ . Então  $X_{1:n} \perp\!\!\!\perp \theta | P_{x_{1:n}}$ , então  $P_{x_{1:n}}$  é uma estatística suficiente.

<sup>11</sup> Teorema Multinomial

$$(x_1 + x_2 + \dots + x_m)^n = \sum_{k_1 + k_2 + \dots + k_m = n} \binom{n}{k_1, k_2, \dots, k_m} \quad (169)$$



## Codificação

O codificador é responsável por associar cada sequência  $x_1, x_2, \dots, x_n$  produzida pela fonte, ou seja, realizações de variáveis aleatórias  $X_1, X_2, \dots, X_n$ , a uma sequência de bits de comprimento variável ou fixo, de modo que a sequência original possa ser recuperada perfeitamente por um decodificador. Para simplificar, vamos supor que as mensagens codificadas possuem, todas elas, um mesmo comprimento  $m$ . O alfabeto da fonte é  $\mathcal{X} = \{a_1, a_2, \dots, a_K\}$ , possuindo assim cardinalidade  $K = |\mathcal{X}|$ . O alfabeto de saída do codificador é  $\mathcal{Y} = \{0, 1\}$ , ou seja, codificação binária ( $|\mathcal{Y}| = 2$ ). A Figura 14 representa este esquema de codificação.

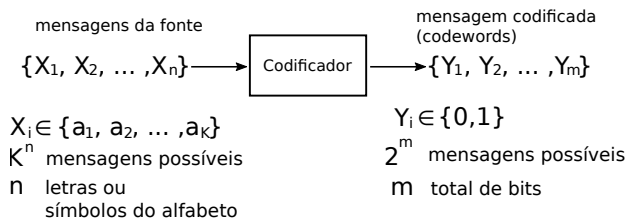


Figura 14: Esquemático de um codificador.

Para que seja possível termos uma palavra de código para cada mensagem possível, devemos satisfazer a seguinte condição:

$$2^m \geq K^n, \quad (170)$$

ou seja,

$$m \geq (\log K)n \quad (171)$$

A taxa de codificação mede a eficiência de um esquema de codificação ao representar uma sequência de símbolos em forma binária.

**Definição 16** (Taxa de Codificação). Para uma sequência de entrada de comprimento  $n$ ,  $X_1, X_2, \dots, X_n$ , codificada em uma mensagem binária de comprimento  $m$ ,  $Y_1, Y_2, \dots, Y_m$ , a taxa de codificação é definida como

$$R = \frac{m}{n}. \quad (172)$$

$R$  representa então o número de bits por símbolo utilizados na tarefa de codificação.

Considerando que a fonte produz uma sequência de símbolos i.i.d., a probabilidade de uma sequência qualquer de comprimento  $n$  pode ser expressa por

$$p(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \prod_{i=1}^n p(X_i = x_i) \quad (173)$$

A informação sobre um evento é dada por  $-\log p(x) = I(x)$ , então a informação associada à mensagem  $x_1, x_2, \dots, x_n$  é

$$I(x_1, x_2, \dots, x_n) = -\log p(x_1, x_2, \dots, x_n) = -\log \prod_{i=1}^n p(x_i) \quad (174a)$$

$$= \sum_{i=1}^n -\log p(x_i) = \sum_{i=1}^n I(x_i), \quad (174b)$$

onde notamos que eventos independentes são aditivos em relação a esta função de informação. Observe que  $EI(X) = H(X)$ . A lei fraca dos grandes números<sup>12</sup> diz que  $\frac{1}{n}S_n \xrightarrow{p} \mu$ , onde  $S_n$  é a soma de v.a.s i.i.d. com média  $\mu = EX_i$ . Temos que  $I(X_i)$  também é uma v.a. com média  $H(X)$ . Obteremos assim

$$\frac{1}{n} \sum_{i=1}^n I(X_i) \xrightarrow[n \rightarrow \infty]{p} H(X). \quad (176)$$

Quando  $n$  é grande suficiente, podemos escrever

$$\frac{1}{n} \sum_{i=1}^n I(x_i) \approx H(X), \forall i, x_i \sim p(x) \quad (177a)$$

$$-\frac{1}{n} \sum_{i=1}^n \log p(x_i) \approx H(X) \quad (177b)$$

$$-\log \prod_{i=1}^n p(x_i) \approx nH(X) \quad (177c)$$

$$-\log p(x_1, x_2, \dots, x_n) \approx nH(X) \quad (177d)$$

$$p(x_1, x_2, \dots, x_n) \approx 2^{-nH(X)}. \quad (177e)$$

Esta probabilidade não depende da sequência em si. Depende apenas do comprimento  $n$  e da entropia da fonte. Quando  $n$  fica grande, podemos dizer que todas as sequências terão a mesma probabilidade:  $2^{-nH}$ . Estas sequências que possuem esta probabilidade (praticamente todas as sequências) são chamadas de *sequências típicas*, e são representadas pelo conjunto  $A_\epsilon^{(n)}$ .

Se todas as sequência de comprimento  $n$  possuem aproximadamente a mesma probabilidade  $2^{-nH(X)}$ , então existe no máximo  $2^{nH}$  sequências de comprimento  $n$ . Pode ser que  $2^{nH} \ll K^n$ , ou seja, o conjunto das sequências típicas é muito menor do que o conjunto de todas as sequências possíveis de comprimento  $n$ . Isto implica que, efetivamente, poderemos nos preocupar apenas com a codificação do conjunto menor, o conjunto das sequências que efetivamente ocorrem, as sequências típicas. Desta forma, para representar (ou codificar) as sequências típicas, precisaremos de  $nH(X)$  bits. Teremos então

$$m = nH(X) \quad (178)$$

no modelo do codificador. Então a taxa será  $H(X)$ .

<sup>12</sup> Lei dos Grandes Números: Se um evento de probabilidade  $p$  é observado repetidamente em ocasiões independentes, a proporção da frequência observada deste evento em relação ao total número de repetições converge em direção a  $p$  à medida que o número de repetições se torna arbitrariamente grande.

Sejam  $X_1, X_2, \dots, X_n$  v.a.s i.i.d. com  $EX_i = \mu$  e  $\text{Var}X_i = \sigma^2 < \infty$ , para  $i = 1, \dots, n$ . Seja a média definida por  $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ , então, para  $\epsilon > 0$ , a *Lei Fraca dos Grandes Números* diz que  $\bar{X}_n$  converge em probabilidade para  $\mu$ , ou seja,

$$\lim_{n \rightarrow \infty} P(|\bar{X}_n - \mu| < \epsilon) = 1. \quad (175)$$

**Teorema 19** (Propriedade da Equipartição Assintótica) *Se  $X_1, X_2, \dots, X_n$  são i.i.d. e  $X_i \sim p(x)$  para todo  $i$ , então*

$$-\frac{1}{n} \log p(X_1, X_2, \dots, X_n) \xrightarrow{p} H(X) \quad (179)$$

*Demonstração.*

$$-\frac{1}{n} \log p(X_1, X_2, \dots, X_n) = -\frac{1}{n} \log \prod_{i=1}^n p(X_i) \quad (180a)$$

$$= -\frac{1}{n} \sum_i \log p(X_i) \xrightarrow{p} E \log p(X) \quad (180b)$$

$$= H(X) \quad (180c)$$

onde utilizamos a lei fraca dos números grandes em 180b. □

**Definição 17** (Conjunto Típico). Um conjunto típico  $A_\epsilon^{(n)}$  em relação a  $p(x)$  é o conjunto de sequências  $(x_1, x_2, \dots, x_n) \in \mathcal{X}^n$  com propriedade

$$2^{-n(H(X)+\epsilon)} \leq p(x_1, x_2, \dots, x_n) \leq 2^{-n(H(X)-\epsilon)} \quad (181)$$

De forma equivalente, podemos escrever

$$A_\epsilon^{(n)} = \left\{ (x_1, x_2, \dots, x_n) : \left| -\frac{1}{n} \log p(x_1, \dots, x_n) - H \right| < \epsilon \right\} \quad (182)$$

**Teorema 20** (Propriedades do Conjunto Típico  $A_\epsilon^{(n)}$ )

1. Se  $(x_1, x_2, \dots, x_n) \in A_\epsilon^{(n)}$ , então

$$H(X) - \epsilon \leq -\frac{1}{n} \log p(x_1, x_2, \dots, x_n) \leq H(X) + \epsilon \quad (183)$$

2.  $p(A_\epsilon^{(n)}) = p\left(\left\{x : x \in A_\epsilon^{(n)}\right\}\right) > 1 - \epsilon$  para  $n$  grande suficiente, para todo  $\epsilon > 0$ .

3. Limite superior:  $|A_\epsilon^{(n)}| \leq 2^{n(H(X)+\epsilon)}$ , onde  $|A_\epsilon^{(n)}|$  é o número de elementos no conjunto  $A_\epsilon^{(n)}$ .

4. Limite inferior:  $|A_\epsilon^{(n)}| \geq (1 - \epsilon)2^{n(H(X)-\epsilon)}$  para  $n$  grande suficiente.

*Demonstração.*

1. O primeiro apenas é uma reformulação da definição de AEP.

2. Vamos utilizar a definição expandida de convergência em probabilidade, dada na equação 179.

$$p(A_\epsilon^{(n)}) = p\left(\left| -\frac{1}{n} \sum_i \log p(x_i) - H \right| < \epsilon\right) > 1 - \delta \quad (184)$$

para  $n$  grande suficiente. Podemos escolher qualquer  $\delta$ , escolhemos então  $\delta = \epsilon$ , resultando em

$$p(A_\epsilon^{(n)}) > 1 - \epsilon, \quad \text{para } n \text{ grande suficiente } \forall \epsilon \quad (185)$$

3. Limite superior de  $A_\epsilon^{(n)}$

$$1 = \sum_x p(x) \geq \sum_{x \in A_\epsilon^{(n)}} p(x) \geq \sum_{x \in A_\epsilon^{(n)}} 2^{-n(H(X)+\epsilon)} \quad (186a)$$

$$= |A_\epsilon^{(n)}| 2^{-n(H(X)+\epsilon)} \quad (186b)$$

Resultando em  $|A_\epsilon^{(n)}| \leq 2^{n(H+\epsilon)}$ .

4. Limite inferior do tamanho de  $A_\epsilon^{(n)}$ . Para  $n$  grande suficiente

$$1 - \epsilon < p(A_\epsilon^{(n)}) \leq \sum_{x \in A_\epsilon^{(n)}} 2^{-n(H(X)-\epsilon)} \quad (187a)$$

$$= 2^{-n(H(X)-\epsilon)} |A_\epsilon^{(n)}| \quad (187b)$$

resultando em  $|A_\epsilon^{(n)}| \geq (1 - \epsilon) 2^{n(H(X)-\epsilon)}$ .

□

A AEP e o tamanho do conjunto típico têm implicações diretas na codificação de sequências, como no exemplo de imagens digitais. Considere uma imagem full HD de  $1080 \times 720$  pixels, com 16 milhões de cores (24 bits por pixel, ou seja,  $2^{24}$  cores possíveis). O número total de pixels é  $1080 \times 720 = 777.600$ , e cada pixel é representado por 24 bits, totalizando  $K = 1080 \times 720 \times 24 = 18.662.400 \approx 10^7$  bits por imagem. O número total de imagens possíveis é  $2^K = 2^{18.662.400} \approx 10^{5.617 \times 10^6}$ , um valor extremamente grande, muito maior que o número de átomos no universo observável ( $\approx 10^{81}$ ). Pela AEP, se os pixels fossem i.i.d. com entropia  $H(X)$  (em bits por pixel), o número de imagens típicas seria aproximadamente  $2^{nH(X)}$ , onde  $n = 777.600$ . Mesmo que  $H(X)$  fosse pequeno (por exemplo, 1 bit por pixel devido a redundâncias), o número de imagens típicas seria  $2^{777.600} \approx 10^{234.000}$ , ainda muito maior que  $10^{81}$ . Isso ilustra que, embora o número de imagens típicas seja uma fração minúscula do total, ele ainda é astronomicamente grande, destacando a necessidade de codificação eficiente para lidar com sequências típicas, que dominam a probabilidade.

Considere agora as chaves privadas de Bitcoin. Uma chave privada de Bitcoin é um número de 256 bits, gerado aleatoriamente, o que significa que o número total de chaves possíveis é  $2^{256} \approx 1,1579 \times 10^{77}$ . Esse valor é menor que o número de átomos no universo observável, mas ainda é astronomicamente grande. Pela AEP, se os bits da chave fossem gerados de forma i.i.d. com entropia  $H(X)$  (em bits por bit),



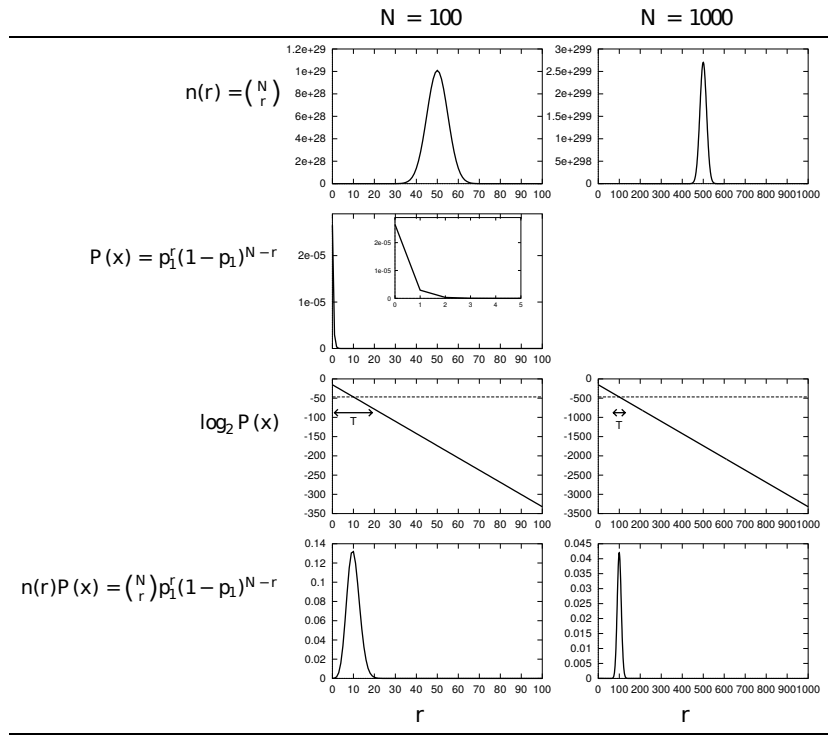


Figura 16: Para  $p = 0.1$ ,  $n = 100$  e  $n = 1000$  os gráficos ilustram  $n(r)$ , o número de strings contendo  $r$  1s; a probabilidade  $P(x_{1:n})$  para uma string contendo  $r$  1s; a mesma probabilidade em escala logarítmica; e a probabilidade total  $n(r)P(x_{1:n})$  de todas as strings contendo  $r$  1s (MacKay 2003).

de codificação considerando a codificação de sequências típicas. Como vimos que existe um limite para o tamanho do conjunto típicos, iremos utilizar o número de bits mínimo necessários para codificar as sequências quer pertençam a este conjunto. As sequências remanescentes, contidas no conjunto não típico, podem ser codificadas utilizando mais bits, sem que isto cause impacto significativo ao código, uma vez que a probabilidade do conjunto típico é aproximadamente 1 (conforme item 2 do Teorema 20).

A ideia consiste em particionar o conjunto de sequências em dois blocos: conjunto típico  $A_\epsilon^{(n)}$ , e o seu complemento, o conjunto não típico  $A_\epsilon^{(n)c} \triangleq \mathcal{X}^n \setminus A_\epsilon^{(n)}$ . Tais partições satisfazem  $A_\epsilon^{(n)} \cap A_\epsilon^{(n)c} = \emptyset$ , e  $A_\epsilon^{(n)} \cup A_\epsilon^{(n)c} = \mathcal{X}^n$ . A Figura 17 ilustra o particionamento proposto.

Vamos então indexar os elementos de cada conjunto (conjunto típico e não-típico) separadamente. O número de elementos do conjunto típico é  $|A_\epsilon^{(n)}| \leq 2^{n(H+\epsilon)}$ , desta forma será necessário

$$\lceil n(H + \epsilon) \rceil \leq n(H + \epsilon) + 1 \text{bits} \quad (188)$$

para indexar os elementos deste conjunto. Ainda, utilizaremos um bit extra para indicar se o elemento está no conjunto típico ou não, i.e., vamos utilizar uma sequência  $(b_0, b_1, b_2, \dots, b_{\lceil n(H+\epsilon) \rceil})$  onde o primeiro bit indica se temos um elemento do conjunto típico ( $b_0 = 0$ ) ou não

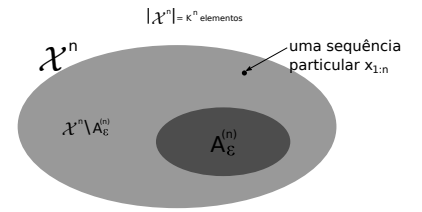


Figura 17: Particionamento de  $\mathcal{X}^n$  em dois conjuntos:  $A_\epsilon^{(n)}$  e  $A_\epsilon^{(n)c}$ .

e os demais indexam o elemento do conjunto. O número total de bits necessário para uma sequência típica será então  $\leq n(H + \epsilon) + 2$ .

Para os elementos do conjunto não típico, vamos utilizar

$$\lceil \log |\mathcal{X}|^n \rceil \leq n \log K + 1 \text{ bits.} \quad (189)$$

Então teremos um vetor binário da forma  $(b_0, b_1, b_2, \dots, b_{\lceil \log |\mathcal{X}|^n \rceil})$  onde  $b_0 = 1$ , indicando a atipicidade. Assim, o número total de bits para indexar uma sequência atípica será  $\leq n \log K + 2$ .

Este código proposto é 1-pra-1, sendo fácil codificar e decodificar, dado o *codebook*. Para o conjunto não-típico  $A_\epsilon^{(n)c}$  estamos utilizando mais bits do que o necessário, já que  $|A_\epsilon^{(n)c}| = |\mathcal{X}^n| - |A_\epsilon^{(n)}| = K^n - |A_\epsilon^{(n)}| \leq K^n$ , mas isto não importará, como veremos adiante. O importante desta proposta de código é que as sequências típicas possuem um comprimento descritivo curto, aproximadamente  $nH$ .

**Definição 18** (Comprimento da palavra associada a uma sequência). O comprimento da palavra (*codeword*) associada à sequência  $x_{1:n}$  é denominado  $l(x_{1:n})$ .

Assim,  $l(X_{1:n})$  é uma variável aleatória, já que  $X_{1:n}$  é uma variável aleatória. Então  $El(X_{1:n}) = \sum_{x_{1:n}} p(x_{1:n})l(x_{1:n})$  é o valor esperado do comprimento do código. Queremos que ele seja o menor possível.

Suponha que  $n$  seja grande suficiente de forma que  $p(A_\epsilon^{(n)}) > 1 - \epsilon$ , então

$$El(X_{1:n}) = \sum_{x_{1:n}} p(x_{1:n})l(x_{1:n}) \quad (190a)$$

$$= \sum_{x_{1:n} \in A_\epsilon^{(n)}} p(x_{1:n})l(x_{1:n}) + \sum_{x_{1:n} \in A_\epsilon^{(n)c}} p(x_{1:n})l(x_{1:n}) \quad (190b)$$

$$\leq \sum_{x_{1:n} \in A_\epsilon^{(n)}} p(x_{1:n})[n(H + \epsilon) + 2] + \sum_{x_{1:n} \in A_\epsilon^{(n)c}} p(x_{1:n})[n \log K + 2] \quad (190c)$$

$$= \underbrace{p(A_\epsilon^{(n)})}_{\leq 1} [n(H + \epsilon) + 2] + \underbrace{p(A_\epsilon^{(n)c)})}_{< \epsilon} [n \log K + 2] \quad (190d)$$

$$\leq n(H + \epsilon) + 2 + \epsilon n \log K + \epsilon 2 \quad (190e)$$

$$= n \underbrace{[H + \epsilon + \epsilon \log K + \frac{2}{n} + \frac{2\epsilon}{n}]}_{\epsilon'} = n(H + \epsilon'), \quad (190f)$$

onde definimos  $\epsilon'$  como

$$\epsilon' = \epsilon + \epsilon \log K + \frac{2}{n} + \frac{2\epsilon}{n}. \quad (191)$$

Podemos fazer  $\epsilon'$  tão pequeno quanto queremos, fazendo  $\epsilon$  pequeno e  $n$  grande. Desta forma, podemos fazer  $n(H + \epsilon')$  tão próximo quanto quisermos de  $nH$ .

**Teorema 21** (Primeiro Teorema de Shannon) *Seja  $X_{1:n}$  i.i.d.  $\sim p(x)$ ,  $\epsilon > 0$ , então  $\exists$  um código  $f_n : \mathcal{X}^n \rightarrow \text{string binária}$  e um inteiro  $n_\epsilon$ , tal que o mapeamento seja um-pra-um (desta forma inversível sem erro), e*

$$E\left[\frac{1}{n}l(X_{1:n})\right] \leq H(X) + \epsilon \quad (192)$$

para todo  $\epsilon > 0$  e todo  $n \geq n_\epsilon$ .

*Demonstração.* A demonstração do Primeiro Teorema de Shannon usa a codificação de sequências típicas de maneira semelhante ao exemplo anterior. □

É interessante notar que a sequência mais provável não pertence ao conjunto típico  $A_\epsilon^{(n)}$ . Isso ocorre porque o conjunto típico contém sequências cujo histograma empírico é próximo da distribuição subjacente. A tipicidade não está relacionada à probabilidade de uma única sequência ser a maior.

Dado que o conjunto típico é tal que  $P(A_\epsilon^{(n)}) \rightarrow 1$  à medida que  $n \rightarrow \infty$ , ou seja,  $A_\epsilon^{(n)}$  captura toda probabilidade, podemos nos questionar se existe um conjunto ainda menor que também contenha essencialmente ‘toda’ a probabilidade.

**Definição 19** (Sequências assintoticamente iguais até a primeira ordem do expoente). A notação  $a_n \doteq b_n$  indica que  $a_n$  e  $b_n$  são iguais até a primeira ordem do expoente, ou seja, suas taxas de crescimento exponencial são as mesmas à medida que  $n \rightarrow \infty$ . Formalmente, isso significa que  $\lim_{n \rightarrow \infty} \frac{1}{n} |\log a_n - \log b_n| = 0$ , de modo que  $\frac{1}{n} \log a_n \rightarrow \alpha$  e  $\frac{1}{n} \log b_n \rightarrow \alpha$  para o mesmo  $\alpha$ . Para  $n$  grande,  $a_n$  e  $b_n$  possuem aproximadamente o mesmo comportamento.

**Teorema 22** *Seja  $X_{1:n}$  uma sequência i.i.d.  $\sim p(x)$ . Para  $\delta < 1/2$  e qualquer  $\delta' > 0$ , se  $P(B_\delta^{(n)}) > 1 - \delta$ , então*

$$\frac{1}{n} \log |B_\delta^{(n)}| > H - \delta', \quad (193)$$

se  $n$  é grande suficiente. Teremos assim

$$|B_\delta^{(n)}| > 2^{n(H-\delta')} \approx 2^{nH} \quad (194)$$

Usando a Definição 19, podemos então reescrever o teorema anterior da seguinte forma:

Se  $\delta_n \rightarrow 0$  e  $\epsilon_n \rightarrow 0$ , então teremos

$$|B_{\delta_n}^{(n)}| \doteq |A_{\epsilon_n}^{(n)}| \doteq 2^{nH}. \quad (195)$$

*Demonstração.* Seja  $X_1, X_2, \dots, X_n$  i.i.d.  $\sim p(x)$ . Seja  $B_\delta^{(n)} \subset \mathcal{X}^n$  tal que  $\Pr(B_\delta^{(n)}) > 1 - \delta$ . Fixe  $\epsilon < 1/2$ . Dados dois subconjuntos quaisquer



$A$  e  $B$  tais que  $\Pr(A) > 1 - \epsilon_1$  e  $\Pr(B) > 1 - \epsilon_2$ . Seja  $A^c$  o complemento de  $A$  e  $B^c$  o complemento de  $B$ , então

$$P(A^c \cup B^c) \leq P(A^c) + P(B^c). \quad (196)$$

Como  $P(A) > 1 - \epsilon_1$ , teremos  $P(A^c) \leq \epsilon_1$ . De forma similar,  $P(B^c) \leq \epsilon_2$ . Poderemos assim escrever

$$P(A \cap B) = 1 - P(A^c \cup B^c) \quad (197a)$$

$$\geq 1 - P(A^c) - P(B^c) \quad (197b)$$

$$\geq 1 - \epsilon_1 - \epsilon_2. \quad (197c)$$

Podemos reescrever a desigualdade anterior como

$$\Pr(A_\epsilon^{(n)} \cap B_\delta^{(n)}) \geq 1 - \epsilon - \delta. \quad (198)$$

A probabilidade de um conjunto é dada pela soma das probabilidades de todos os elementos (sequências) neste conjunto, logo teremos

$$\Pr(A_\epsilon^{(n)} \cap B_\delta^{(n)}) = \sum_{x^n \in A_\epsilon^{(n)} \cap B_\delta^{(n)}} p(x^n) \quad (199)$$

A probabilidade dos elementos no conjunto típico é limitada por  $2^{-n(H-\epsilon)}$ .

Desta forma teremos

$$1 - \epsilon - \delta \leq \Pr(A_\epsilon^{(n)} \cap B_\delta^{(n)}) \quad (200a)$$

$$= \sum_{x^n \in A_\epsilon^{(n)} \cap B_\delta^{(n)}} p(x^n) \quad (200b)$$

$$\leq \sum_{x^n \in A_\epsilon^{(n)} \cap B_\delta^{(n)}} 2^{-n(H-\epsilon)} \quad (200c)$$

$$= |A_\epsilon^{(n)} \cap B_\delta^{(n)}| 2^{-n(H-\epsilon)} \quad (200d)$$

$$\leq |B_\delta^{(n)}| 2^{-n(H-\epsilon)}, \quad (200e)$$

onde utilizamos  $A_\epsilon^{(n)} \cap B_\delta^{(n)} \subseteq B_\delta^{(n)}$ .

Poderemos reescrever então da seguinte forma,

$$|B_\delta^{(n)}| > 2^{n(H-\epsilon)}, \quad (201)$$

onde  $\epsilon > 0$ .

□

### Método de Tipos

O método de tipos refina a abordagem das sequências típicas, oferecendo uma análise mais estruturada das propriedades assintóticas. O método de tipos considera todas as possíveis distribuições empíricas

(ou ‘tipos’) que sequências  $x_{1:n}$  podem ter, agrupando-as em conjuntos de sequências que compartilham um mesmo histograma empírico  $\hat{p}_n(x_{1:n})$ , ou  $P_{x_{1:n}}$ . O conjunto de sequências de comprimento  $n$  pode ser particionado em subconjuntos de sequências de tipos distintos. Na AEP, o método de tipos confirma que o número de sequências típicas é  $\doteq 2^{nH(X)}$ . Vamos então estabelecer algumas definições.

**Definição 20** (Tipo). Seja  $X_1, X_2, \dots, X_n \equiv X_{1:n}$  uma amostra de comprimento  $n$  de uma variável aleatória discreta  $D$ -ária. Então  $x_i \in \mathcal{X}$  e o tamanho do alfabeto é  $D = |\mathcal{X}|$ , e  $\mathcal{X} = \{a_1, a_2, \dots, a_D\}$ . Definimos a seguinte estatística, o histograma empírico da amostras, também chamado tipo da amostra:

$$P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \dots, \frac{n(a_D|x_{1:n})}{n} \right) \quad (202)$$

onde  $n(a_i|x_{1:n})$  representa o número de ocorrências do símbolo  $a_i$  na amostra  $x_{1:n}$ .

Note que  $P_{x_{1:n}}$  pode ser considerado como uma função massa probabilística. Um tipo  $\hat{p}$  é uma função de massa de probabilidade sobre  $S_X$ , definido como o histograma empírico  $\hat{p}(a) = \frac{k_a}{n}$ , onde  $k_a = \sum_{i=1}^n \mathbb{I}\{x_i = a\}$  é o número de ocorrências do símbolo  $a \in S_X$  na sequência, e  $\sum_{a \in S_X} k_a = n$ .

**Definição 21** (Conjunto de Tipos). O Conjunto de Tipos  $\mathcal{P}_n$ , ou  $\mathcal{P}_n(\mathcal{X})$ , para sequências de comprimento  $n$  sobre um alfabeto finito  $\mathcal{X}$  é o conjunto de todas as possíveis distribuições empíricas (ou tipos) que podem ser formadas por sequências  $(x_1, x_2, \dots, x_n) \in \mathcal{X}^n$ .

**Exemplo 9** (Ensaio de Bernoulli). Para sequências de comprimento  $n$  geradas a partir de um ensaio de Bernoulli com alfabeto  $\mathcal{X} = \{0, 1\}$ , teremos o seguinte conjunto de tipos:

$$\mathcal{P}_n(\mathcal{X}) = \left\{ \left( \frac{0}{n}, \frac{n}{n} \right), \left( \frac{1}{n}, \frac{n-1}{n} \right), \dots, \left( \frac{n}{n}, \frac{0}{n} \right) \right\} \quad (203)$$

onde existem  $n + 1$  tipos (histogramas empíricos). O primeiro tipo em 203 representa a sequência sem nenhuma ocorrência de zeros e  $n$  ocorrências de uns; o segundo tipo representa as sequências com apenas uma ocorrência de zero e  $n - 1$  ocorrência de uns; e o último tipo representa a sequência com  $n$  ocorrência de zeros e nenhuma ocorrência de uns.

**Definição 22** (Classe de Tipo). A Classe de Tipo  $T(P)$ , ou  $T(\hat{p})$ , associada a um tipo  $\hat{p} \in \mathcal{P}_n(\mathcal{X})$ , para sequências de comprimento  $n$  sobre um alfabeto finito  $\mathcal{X}$ , é o conjunto de todas as sequências  $(x_1, x_2, \dots, x_n) \in \mathcal{X}^n$  que compartilham um mesmo histograma empírico  $\hat{p}$ , ou  $P$ .

$$T(P) \triangleq \{x_{1:n} \in \mathcal{X}^n : P_{x_{1:n}} = P\}. \quad (204)$$

**Exemplo 10** (Ensaio de Bernoulli). Para o caso do ensaio de Bernoulli, vejamos alguns exemplos de classe de tipo.

Para o tipo  $P = (\frac{0}{n}, \frac{n}{n})$ , a classe de tipo  $T(P)$  possui apenas uma sequência:

$$T(P) = \{111 \cdots 11\}. \quad (205)$$

Para o tipo  $P = (\frac{1}{n}, \frac{n-1}{n})$ , a classe de tipo  $T(P)$  possui  $n$  sequências:

$$T(P) = \{011 \cdots 11, 101 \cdots 11, \dots, 111 \cdots 01, 111 \cdots 10\}. \quad (206)$$

Para o tipo  $P = (\frac{n}{n}, \frac{0}{n})$ , a classe de tipo  $T(P)$  possui apenas uma sequência:

$$T(P) = \{000 \cdots 00\}. \quad (207)$$

Como mencionado anteriormente, o conjunto de todas sequências  $\mathcal{X}^n$  pode ser particionado em diferentes conjuntos  $T(P_i)$ , com  $P_i \in \mathcal{P}_n$ , o conjunto de todos os tipos, ou seja,  $\mathcal{P}_n = \{P_1, P_2, \dots, P_{|\mathcal{P}_n|}\}$ . As partições são disjuntas

$$T(P_i) \cap T(P_j) = \emptyset, \forall i, j \in \{1, \dots, |\mathcal{P}_n|\} \text{ e } i \neq j, \quad (208)$$

e o particionamento é exaustivo

$$\bigcup_{P \in \mathcal{P}_n} T(P) = \mathcal{X}^n. \quad (209)$$

Este particionamento é ilustrado na Figura 18.

No exemplo do ensaio de Bernoulli é fácil constatar quantos tipos distintos existem. O Teorema 24 a seguir usa o método estrela traço da análise combinatória para determinar o número de tipos existentes. Vejamos então primeiramente o teorema da análise combinatória.

**Teorema 23** (Método Estrela Traço) *Suponha que  $n$  estrelas que devem ser organizadas em  $k$  recipientes, podendo inclusive ter recipiente vazio. Neste caso, o número de maneiras de fazer tal organização é dado por*

$$\binom{n+k-1}{k-1} \quad (210)$$

*Demonstração.* Este problema é o mesmo que incluir  $k-1$  traços para separar uma sequência de  $n$  estrelas. Os traços podem ser inseridos em qualquer posição, inclusive sem existir estrelas entre eles, como por exemplo, com  $n=7$  e  $k=4$ :

$$\star \star \star \star \quad | \quad | \quad \star \quad | \quad \star \star$$

Note que, neste problema temos  $n+k-1$  símbolos (estrelas e traços) para serem incluídos, dos quais  $k-1$  são escolhidos para serem traços. Desta forma existem  $\binom{n+k-1}{k-1}$  formas de dispor estes símbolos.

□

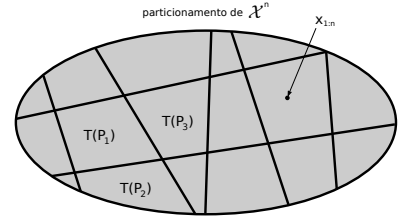


Figura 18: Particionamento do espaço de sequências de comprimento  $n$ .

**Teorema 24** (Número de tipos existentes) *Para sequências de comprimento  $n$ , em um alfabeto  $\mathcal{X}$ , o número de tipos é dado por*

$$|\mathcal{P}_n| = \binom{n + |\mathcal{X}| - 1}{|\mathcal{X}| - 1} \quad (211)$$

*Demonstração.* Um histograma empírico pode ser visto como uma variação do problema estrela-traço. Se temos um alfabeto com  $|\mathcal{X}| = D$  símbolos, o Conjunto de Tipos  $\mathcal{P}_n$  contém todas as possíveis distribuições empíricas  $\hat{p}$  para sequências de comprimento  $n$  sobre o alfabeto  $\mathcal{X}$ . Um tipo  $\hat{p}$  é uma função de massa de probabilidade tal que  $\hat{p}(a) = k_a/n$ , onde  $k_a$  é o número de ocorrências do símbolo  $a \in \mathcal{X}$ , e  $\sum_{a \in \mathcal{X}} k_a = n$ . Assim,  $k_a \in \{0, 1, \dots, n\}$ , e a restrição é que a soma de todos  $k_a$ 's deve ser igual a  $n$ , o comprimento da sequência. Cada tipo  $\hat{p}$  é definido pela ênupla  $(k_{a_1}, k_{a_2}, \dots, k_{a_D})$ , e cada uma delas corresponde a uma forma de organização no problema de estrelas e traços.  $\square$

Embora tenhamos uma fórmula fechada para o número de tipos, Equação (211), iremos encontrar um limite superior para  $|\mathcal{P}_n|$ , que, embora seja um limite relativamente largo, é mais fácil de manipular e suficiente para muitas demonstrações teóricas.

**Teorema 25** (Limite no número de tipos) *O número de tipos para sequências de comprimento  $n$  em um alfabeto  $\mathcal{X}$  é limitado por*

$$|\mathcal{P}_n| \leq (n + 1)^{|\mathcal{X}|}. \quad (212)$$

*Demonstração.* O numerador de cada entrada em um tipo pode assumir  $(n + 1)$  valores distintos (de 0 a  $n$ ). Existem  $|\mathcal{X}|$  entradas em um tipo, e portanto a mesma quantidade de numeradores. Os valores dos numeradores interagem entre si (a soma de todos deve ser igual a  $n$ ), mas podemos achar um limite superior desconsiderando esta interação.

$$|\mathcal{P}_n| \leq \underbrace{(n + 1) \times (n + 1) \times \dots \times (n + 1)}_{|\mathcal{X}| \text{ vezes}} = (n + 1)^{|\mathcal{X}|}. \quad (213)$$

$\square$

É importante notar na Equação (212) que existe no máximo um número polinomial em  $n$  de tipos de sequências de comprimento  $n$ . Entretanto, sabemos que o número de sequências cresce com  $n$ ,  $|\mathcal{X}|^n$ . Com  $n$  grande suficiente, eventualmente, teremos que apenas um único tipo (o tipo das sequências típicas) conterà todas as sequências, como será demonstrado no Teorema 32.

Para sequências geradas por uma fonte i.i.d., a probabilidade de uma sequência  $(x_1, x_2, \dots, x_n)$  depende apenas do seu tipo e não da sequência específica. Isso significa que todas as sequências em uma mesma classe de tipo, ou seja, sequências que compartilham um mesmo histograma empírico, têm a mesma probabilidade.

**Teorema 26** (Probabilidade Depende do Tipo) *Seja  $X_1, X_2, \dots, X_n$  i.i.d.  $\sim Q(x)$ , com  $Q$  arbitrário, e extensão  $Q^n(x_{1:n}) = \prod_i Q(x_i)$ , a probabilidade da sequência depende apenas do tipo, ou seja, a probabilidade é ‘independente’ da sequência, dado o tipo e  $Q$ , isto é,*

$$Q^n(x_{1:n}) = 2^{-n[H(P_{x_{1:n}}) + D(P_{x_{1:n}} || Q)]}. \quad (214)$$

*Demonstração.*

$$Q^n(x_{1:n}) = \prod_{i=1}^n Q(x_i) = \prod_{a \in \mathcal{X}} Q(a)^{n(a|x_{1:n})} \quad (215a)$$

$$= \prod_{a \in \mathcal{X}} Q(a)^{nP_{x_{1:n}}(a)} = \prod_{a \in \mathcal{X}} 2^{\{nP_{x_{1:n}}(a) \log Q(a)\}} \quad (215b)$$

$$= \prod_{a \in \mathcal{X}} 2^{\left\{ n \left[ P_{x_{1:n}}(a) \log Q(a) - \underbrace{P_{x_{1:n}}(a) \log P_{x_{1:n}}(a) + P_{x_{1:n}}(a) \log P_{x_{1:n}}(a)}_{=0} \right] \right\}} \quad (215c)$$

$$= 2^{\sum_{a \in \mathcal{X}} \left( -P_{x_{1:n}}(a) \log \frac{P_{x_{1:n}}(a)}{Q(a)} + P_{x_{1:n}}(a) \log P_{x_{1:n}}(a) \right)} \quad (215d)$$

$$= 2^{-n(D(P_{x_{1:n}} || Q) + H(P_{x_{1:n}}))}. \quad (215e)$$

□

Se  $Q$  é uma distribuição racional (i.e., um tipo possível) e se  $x_{1:n} \in T(Q)$ , então

$$Q^n(x_{1:n}) = 2^{-nH(Q)}. \quad (216)$$

Se  $Q$  for irracional, podemos fazer  $D(P_{x_{1:n}} || Q)$  tão pequeno quando desejável, fazendo  $n$  grande suficiente.

No método de tipos, uma questão natural é identificar qual classe de tipo tem a maior probabilidade total quando as sequências são geradas por uma determinada distribuição. Intuitivamente, esperamos que a classe de tipo mais provável seja aquela cujo histograma empírico esteja mais próximo da distribuição geradora.

**Lema 27** (Classe de Tipo com maior probabilidade) *Dada a distribuição geradora  $Q = P \in \mathcal{P}_n$ , teremos que  $T(P)$  possui a maior probabilidade. Isto é*

$$P^n(T(P)) \geq P^n(T(\hat{P})), \quad \forall \hat{P} \in \mathcal{P}_n. \quad (217)$$

*Demonstração.*

$$\frac{P^n(T(P))}{P^n(T(\hat{P}))} = \frac{|T(P)| \prod_{a \in \mathcal{X}} P(a)^{nP(a)}}{|T(\hat{P})| \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)}} \quad (218a)$$

$$= \frac{\binom{n}{nP(a_1) \ nP(a_2) \ \dots \ nP(a_D)}}{\binom{n}{n\hat{P}(a_1) \ n\hat{P}(a_2) \ \dots \ n\hat{P}(a_D)}} \prod_{a \in \mathcal{X}} P(a)^{nP(a)} \quad (218b)$$

$$= \prod_{a \in \mathcal{X}} \frac{[n\hat{P}(a)]!}{[nP(a)]!} P(a)^{n(P(a) - \hat{P}(a))} \quad (218c)$$

$$\geq \prod_{a \in \mathcal{X}} (nP(a))^{n(\hat{P}(a) - P(a))} P(a)^{n(P(a) - \hat{P}(a))} \quad (218d)$$

$$= \prod_{a \in \mathcal{X}} n^{n(\hat{P}(a) - P(a))} \quad (218e)$$

$$\geq \prod_{a \in \mathcal{X}} n^{n(\hat{P}(a) - P(a))} \quad (218f)$$

$$= n^{n[\sum_{a \in \mathcal{X}} \hat{P}(a) - \sum_{a \in \mathcal{X}} P(a)]} \quad (218g)$$

$$= n^{n(1-1)} = 1, \quad (218h)$$

onde em 218d utilizamos que  $\frac{m!}{n!} \geq n^{m-n}$ , para  $m$  e  $n$  inteiros não negativos<sup>13</sup>. Por fim, temos que  $P^n(T(P)) \geq P^n(T(\hat{P}))$ . □

O tamanho de uma classe de tipo  $T(P)$  pode ser determinado pelos coeficientes multinomiais. Cada sequência  $x_{1:n} \in T(P)$  possui exatamente  $nP(a)$  ocorrências de cada símbolo  $a \in \mathcal{X}$ . O número de sequências corresponde ao número de maneiras de permutar os símbolos de acordo com o número de ocorrência de cada símbolo.

**Lema 28** (Tamanho da Classe de Tipo) *Seja  $P \in \mathcal{P}_n$ , um tipo para sequências de comprimento  $n$  sobre um alfabeto finito  $\mathcal{X}$ , de tamanho  $D = |\mathcal{X}|$ , e seja  $T(P)$  a classe de tipo associada, ou seja, o conjunto de todas as sequências  $x_{1:n} \in \mathcal{X}^n$  com histograma empírico  $P$ . Então o tamanho de  $T(P)$  é dado por*

$$|T(P)| = \binom{n}{nP(a_1) \ nP(a_2) \ \dots \ nP(a_D)} = \frac{n!}{\prod_{a \in \mathcal{X}} (nP(a))!}. \quad (219)$$

*Demonstração.* Cada sequência em  $T(P)$  tem exatamente  $nP(a)$  ocorrências de cada símbolo  $a \in \mathcal{X}$ , com  $\sum_{a \in \mathcal{X}} (nP(a)) = n$ . O número de sequências distintas é o número de maneiras de permutar os  $n$  símbolos, considerando que as permutações dentro de cada símbolo não geram sequências distintas. Assim, o número de permutações distintas é o coeficiente multinomial, como dado na Equação (219). □

Novamente, apesar de termos o resultado exato, dado pela Equação (219), em muitas situações é mais prático utilizarmos limites que

<sup>13</sup> Sejam  $m$  e  $n$  inteiros não negativos, então  $\frac{m!}{n!} \geq n^{m-n}$ . Se  $m > n$ , então  $\frac{m!}{n!} = m(m-1) \dots (n+1) \geq n^{m-n}$ . Se  $m < n$ , então  $\frac{m!}{n!} = \frac{1}{n(n-1) \dots (m+1)} \geq \frac{1}{n^{n-m}}$ . Se  $m = n$ ,  $\frac{m!}{n!} = 1 = n^0$ .

nos forneçam uma notação mais intuitiva e prática para utilização em outras demonstrações. Veremos então a seguir os limites superior e inferior para o tamanho da classe de tipo.

**Teorema 29** (Limites no tamanho da Classe de Tipo) *Dado um tipo  $P \in \mathcal{P}_n$ , temos*

$$\frac{1}{(n+1)^{|\mathcal{X}|}} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)}. \quad (220)$$

*Demonstração (limite superior).*

$$1 \geq P^n(T(P)) = \sum_{x_{1:n} \in T(P)} P^n(x_{1:n}) = \sum_{x_{1:n} \in T(P)} 2^{-nH(P)} \quad (221a)$$

$$= |T(P)| 2^{-nH(P)} \quad (221b)$$

□

*Demonstração (limite inferior).*

$$1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \leq \sum_{Q \in \mathcal{P}_n} \max_{R \in \mathcal{P}_n} P^n(T(R)) \quad (222a)$$

$$= \sum_{Q \in \mathcal{P}_n} P^n(T(P)) \leq (n+1)^{|\mathcal{X}|} P^n(T(P)) \quad (222b)$$

$$= (n+1)^{|\mathcal{X}|} \sum_{x_{1:n} \in T(P)} P^n(x_{1:n}) \quad (222c)$$

$$= (n+1)^{|\mathcal{X}|} \sum_{x_{1:n} \in T(P)} 2^{-nH(P)} \quad (222d)$$

$$\leq (n+1)^{|\mathcal{X}|} \sum_{x_{1:n} \in T(P)} 2^{-nH(P)} \quad (222e)$$

$$= (n+1)^{|\mathcal{X}|} |T(P)| 2^{-nH(P)}, \quad (222f)$$

onde em 222b utilizamos

$$P = \operatorname{argmax}_{R \in \mathcal{P}_n} P^n(T(R)). \quad (223)$$

Ao final, obtemos

$$|T(P)| \geq \frac{1}{(n+1)^{|\mathcal{X}|}} 2^{nH(P)}. \quad (224)$$

□

Para o caso binário,  $\mathcal{X} = \{0, 1\}$ , o Teorema 29 nos fornece os seguintes limites para o tamanho da classe de tipo:

$$\frac{1}{(n+1)^2} 2^{nH(\frac{k}{n})} \leq \binom{n}{k} \leq 2^{nH(\frac{k}{n})}, \quad (225)$$

entretanto, o limite inferior pode ser ainda mais restrito, fornecendo assim

$$\frac{1}{(n+1)} 2^{nH(\frac{k}{n})} \leq \binom{n}{k} \leq 2^{nH(\frac{k}{n})}. \quad (226)$$

Os limites superior e inferior para o tamanho de uma classe de tipo  $T(P)$ , dados no Teorema 29, podem ser utilizados para derivar limites correspondentes na probabilidade total da classe de tipo  $Q^n(T(P))$ , onde  $Q$  é a distribuição subjacente da fonte i.i.d. que gera as sequências. Um resultado importante é que qualquer outro tipo que seja menos próximo de  $Q$  (em termos de divergência de Kullback-Leibler) terá sua probabilidade decrescendo exponencialmente com  $n$ , decrescendo assim mais rapidamente que o tipo mais provável.

**Teorema 30** (Limites da probabilidade da classe de tipo) *Para qualquer  $P \in \mathcal{P}_n$  e seja  $Q$  a distribuição subjacente da fonte i.i.d., a probabilidade da classe de tipo  $T(P)$  sob  $Q^n$  é tal que  $Q^n(T(P)) \doteq 2^{-nD(P||Q)}$ . Especificamente, temos os limites*

$$\frac{1}{(n+1)^{|\mathcal{X}|}} 2^{-nD(P||Q)} \leq Q^n(T(P)) \leq 2^{-nD(P||Q)}. \quad (227)$$

*Demonstração.*

$$Q^n(T(P)) = \sum_{x_{1:n} \in T(P)} Q^n(x_{1:n}) = \sum_{x_{1:n} \in T(P)} 2^{-n(D(P||Q)+H(P))} \quad (228a)$$

$$= |T(P)| 2^{-n(D(P||Q)+H(P))} \quad (228b)$$

para completar a demonstração, os limites do tamanho da classe de tipo dados pela Equação (220). □

Iremos agora rever a definição de conjunto típico, utilizando a formalização do métodos de tipos.

**Definição 23** (Conjunto Típico). Seja  $X_1, X_2, \dots, X_n$  i.i.d.  $\forall i, X_i \sim Q(x)$ . Então o conjunto típico é definido como

$$T_Q^\epsilon = \{x_{1:n} : D(P_{x_{1:n}} || Q) \leq \epsilon\}. \quad (229)$$

Podemos agora analisar a probabilidade do conjunto típico.

**Teorema 31** (Probabilidade do Conjunto Típico) *Sejam  $X_1, X_2, \dots, X_n$  i.i.d.  $\forall i, X_i \sim Q(x)$ . A probabilidade do complemento do conjunto típico  $\bar{T}_Q^\epsilon$  é dada por*

$$Q(\bar{T}_Q^\epsilon) = Q(\{x_{1:n} : D(P_{x_{1:n}} || Q) > \epsilon\}) \leq 2^{-n(\epsilon - |\mathcal{X}| \frac{\log(n+1)}{n})}. \quad (230)$$

Dessa forma,

$$D(P_{x_{1:n}} || Q) \xrightarrow{p} 0, \text{ quando } n \rightarrow \infty, \quad (231)$$

ou seja, a divergência entre  $P_{x_{1:n}}$  e  $Q$  converge em probabilidade para zero quando  $n$  é grande suficiente.



*Demonstração.*

$$Q(\bar{T}_Q^\epsilon) = \sum_{P \in \mathcal{P}_n: D(P||Q) > \epsilon} Q^n(T(P)) \quad (232a)$$

$$\leq \sum_{P \in \mathcal{P}_n: D(P||Q) > \epsilon} 2^{-nD(P||Q)} \quad (232b)$$

$$\leq \sum_{P \in \mathcal{P}_n: D(P||Q) > \epsilon} 2^{-n\epsilon} \quad (232c)$$

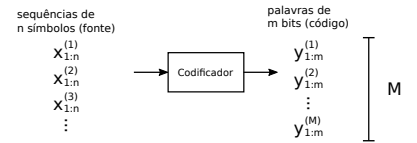
$$\leq (n+1)^{|\mathcal{X}|} 2^{-n\epsilon} = 2^{-n(\epsilon - |\mathcal{X}| \frac{\log(n+1)}{n})} \quad (232d)$$

então a probabilidade vai para zero quando  $n \rightarrow \infty$ , e desta forma a probabilidade do conjunto típico vai para 1 quando  $n \rightarrow \infty$ .  $\square$

Os tipos que divergem (KL) mais do que  $\epsilon$  da distribuição subjacente  $Q$  terão probabilidade decrescente. Para  $n$  grande, o conjunto típico acaba sendo a única coisa que ocorre com uma probabilidade não evanescente.

Dizemos que uma codificação de fonte é universal quando ela não depende da distribuição fonte. Seria possível criar um código universal que atinja o limite da entropia, ou seja, uma taxa  $R > H(Q)$  (em bits por símbolo), onde  $H(Q)$  é a entropia da distribuição subjacente  $Q$ ? O método de tipos oferece uma abordagem para formalizar o teorema de Shannon, permitindo uma análise mais refinada das sequências com base em seus histogramas empíricos. Existem no máximo  $2^{nH(P)}$  sequências do tipo  $P$ . Podermos utilizar  $nH(P)$  bits para representar tais sequências. Se  $R > H(P)$ , podemos utilizar  $nR$  bits para representar estas sequências. Quando  $n$  cresce, apenas os tipos  $P$  ‘próximos’ de  $Q$  irão ocorrer.

A Figura 19 ilustra um codificador de blocos que associa cada uma das  $M$  sequências de  $n$  de símbolos da fonte a uma sequência binária de comprimento  $m$ . Neste codificador de blocos, cada sequência de  $n$  símbolos é codificada conjuntamente associando a ela uma palavra a cada. O codificador faz o mapeamento de sequências de tamanho  $n$  produzidas pela fonte em sequências de  $m$  bits.  $M$  é o número de possíveis mensagens e também o número de palavras do *codebook* do codificador.



**Definição 24** (Código de Bloco com Taxa Fixa  $R$ ). Seja  $X_1, X_2, \dots, X_n \sim Q$ , i.i.d. mas  $Q$  desconhecido. A função do codificador e decodificador são definidas a seguir:

$$\text{codificador: } f_n : \mathcal{X}^n \rightarrow \{1, 2, \dots, 2^{nR}\} \quad (233)$$

$$\text{decodificador: } \phi_n : \{1, 2, \dots, 2^{nR}\} \rightarrow \mathcal{X}^n \quad (234)$$

e a probabilidade de erro

$$P_e^{(n)} = Q^n(\{x_{1:n} : \phi(f_n(x_{1:n})) \neq x_{1:n}\}). \quad (235)$$

Figura 19: Codificador  $(M, n)$ , onde  $M$  representa o número de mensagens e  $n$  o comprimento das sequências.

**Definição 25** (Código de Bloco Universal de Taxa  $R$ ). Um código de bloco de taxa  $R$  para um fonte é dito universal se a função  $f_n$  e  $\phi_n$  não depender da distribuição  $Q$  e se

$$P_e^{(n)} \rightarrow 0 \text{ quando } n \rightarrow \infty \text{ sempre que } H(Q) < R. \quad (236)$$

Veremos que, se  $R > H(Q)$ , então existe uma sequência (em  $n$ ) de códigos com erro evanescente. Por outro lado, se  $R < H(Q)$  a probabilidade de erro vai pra 1.

Na demonstração do Teorema 32 utilizaremos ainda a definição de simplex probabilístico.

**Definição 26** (Simplex Probabilístico). O Simplex Probabilístico em  $\mathbb{R}^m$  é o conjunto de pontos  $x_{1:m} = (x_1, x_2, \dots, x_m) \in \mathbb{R}^m$  tal que  $x_i \geq 0$ ,  $\sum_{i=1}^m x_i = 1$ .

**Exemplo 11** (Simplex Probabilístico com  $m = 2$ ). O Simplex probabilístico será o conjunto de pontos

$$\{(x_1, x_2) : x_1 \geq 0, x_2 \geq 0, x_1 + x_2 = 1\}, \quad (237)$$

representado na Figura 20.

**Exemplo 12** (Simplex Probabilístico com  $m = 3$ ). O Simplex probabilístico será o conjunto de pontos

$$\{(x_1, x_2, x_3) : x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_1 + x_2 + x_3 = 1\} \quad (238)$$

representado na Figura 21 e com alguns pontos em destaque na Figura 22.

O Teorema da Codificação de Shannon é um dos resultados fundamentais da teoria da informação, estabelecendo as condições sob as quais é possível representar as informações produzidas por uma fonte i.i.d. com distribuição subjacente  $Q$  sem que haja perdas, utilizando, para tanto uma taxa  $R$ . O Teorema mostra que, se  $R > H(Q)$ , então existe um código de bloco universal para representar a informação produzida pela fonte com probabilidade de erro tendendo a zero à medida que o comprimento  $n$  das sequências aumenta. Por outro lado, se  $R < H(Q)$ , não é possível representar a informação da fonte com probabilidade de erro evanescente. No contexto do método de tipos, podemos formalizar esse teorema explorando a estrutura das classes de tipo: como o número de tipos é polinomial em  $n$ , enquanto o número de sequências é exponencial em  $n$ , e apenas os tipos próximos de  $Q$  ocorrem, para  $n$  grande, é possível construir códigos universais que codifiquem eficientemente as sequências produzidas pela fonte, usando aproximadamente  $nH(Q)$  bits.

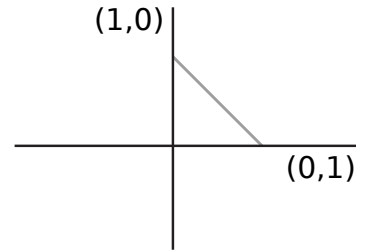


Figura 20: Simplex probabilístico para  $m = 2$ .

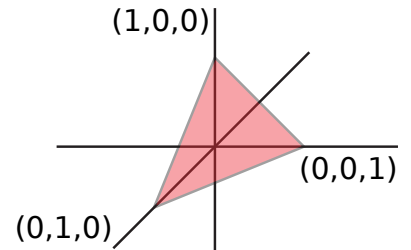


Figura 21: Simplex probabilístico para  $m = 3$ .

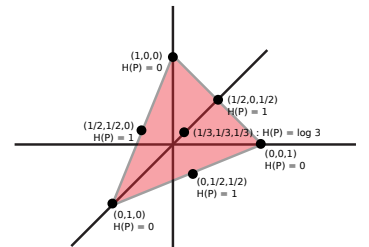


Figura 22: Representação de alguns pontos e suas respectivas entropias.

**Teorema 32** (Teorema da Codificação de Shannon)  $\exists$  uma sequência  $(2^{nR}, n)$  de códigos universais tais que  $P_e^{(n)} \rightarrow 0$  para toda distribuição  $Q$  tal que  $H(Q) < R$ .

*Demonstração.* Fixe  $R > H(Q)$ . Defina uma taxa para  $n$  que é fixada a um fator polinomial:

$$R_n \triangleq R - |\mathcal{X}| \frac{\log(n+1)}{n} < R. \quad (239)$$

Defina um conjunto de sequências que possuem entropia de tipo menor do que esta taxa:

$$A_n \triangleq \{x_{1:n} \in \mathcal{X}^n : H(P_{x_{1:n}}) \leq R_n\} \quad (240a)$$

$$= \left\{ \bigcup_{P \in \mathcal{P}_n} T(P) : H(P) \leq R_n \right\} \quad (240b)$$

A partir desta definição, teremos que

$$|A_n| = \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} |T(P)| \leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nH(P)} \quad (241a)$$

$$\leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nR_n} \leq (n+1)^{|\mathcal{X}|} 2^{nR_n} \quad (241b)$$

$$= 2^{n(R_n + |\mathcal{X}| \frac{\log(n+1)}{n})} = 2^{nR}. \quad (241c)$$

Como  $|A_n| \leq 2^{nR}$ , podemos indexar  $A_n$  com  $nR$  bits.

O codificador será dado por

$$f_n(x_{1:n}) = \begin{cases} \text{índice de } x_{1:n} \text{ em } A_n, & \text{se } x_{1:n} \in A_n \\ 0, & \text{caso contrário.} \end{cases} \quad (242)$$

O codificador associará um índice a  $x_{1:n}$  se  $H(P_{x_{1:n}}) \leq R_n$  (ou seja,  $x_{1:n} \in A_n$ ); e não associará valor se  $H(P_{x_{1:n}}) > R_n$  (ou seja,  $x_{1:n} \notin A_n$ ). Note que  $f_n(\cdot)$  não depende da distribuição da fonte, apenas do ordenamento e de  $\mathbb{R}^m$ . Um erro ocorrerá se  $x_{1:n} \notin A_n$ . Os tipos podem ser representados por pontos em um Simplex Probabilístico, conforme ilustrado na Figura 23.

Um erro ocorre quando a sequência não está em  $A_n$ . Desta forma,

$$P_e^{(n)} = 1 - Q^n(A_n) = Q^n(A_n^c) = \sum_{P: H(P) > R_n} Q^n(T(P)) \quad (243a)$$

$$\leq \sum_{P: H(P) > R_n} \max_{P: H(P) > R_n} Q^n(T(P)) \quad (243b)$$

$$\leq (n+1)^{|\mathcal{X}|} \max_{P: H(P) > R_n} Q^n(T(P)) \quad (243c)$$

$$\leq (n+1)^{|\mathcal{X}|} \max_{P: H(P) > R_n} 2^{-nD(P||Q)} \quad (243d)$$

$$= (n+1)^{|\mathcal{X}|} 2^{-n[\min_{P: H(P) > R_n} D(P||Q)]} \quad (243e)$$

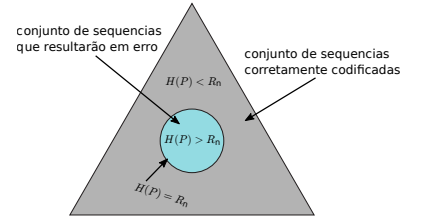


Figura 23: Representação das sequências em um simplex probabilísticos. As sequências com  $H(P_{x_{1:n}}) \leq R_n$  não acarretarão em erro no processo de codificação.

onde utilizamos que  $Q^n(T(P)) \leq 2^{-nD(P||Q)}$ . Temos que  $R_n$  forma uma sequência crescente com  $n$ , tal que  $R_n < R$  para todo  $n$  (veja a Figura 24). Por hipótese,  $H(Q) < R$ . Eventualmente, para algum  $n_0$ , teremos que  $\forall n > n_0, R_n > H(Q)$ . Na Equação (243e), escolhemos  $P : H(P) > R_n$ .

Teremos então:  $H(P) > R_n > H(Q)$ , o que implica em  $P \neq Q$ . Desta forma, teremos  $D(P || Q) > 0$ , para  $P$  que foi escolhido em 243e. Teremos assim

$$P_e^{(n)} \leq \underbrace{(n+1)^{|\mathcal{X}|}}_{\text{polinomial em } n} \underbrace{2^{-n[\min_{P: H(P) > R_n} D(P||Q)]}}_{\text{exp. decrescente qnd } n \rightarrow \infty} \quad (244)$$

Logo,  $P_e^{(n)} \rightarrow 0$  quando  $n \rightarrow \infty$ .

□

Por outro lado, se  $R < H(Q)$  teremos  $P_e^{(n)} \rightarrow 1$ . A entropia é então o limite de representação, ou compressão.

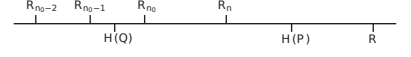


Figura 24: Sequência de taxas  $R_n$ .

# Processos Estocásticos

Até o momento, nossa análise considerou variáveis aleatórias independentes, uma simplificação que facilitou a compreensão de conceitos fundamentais como entropia e informação mútua. Ao final, chegamos à conclusão de que  $nH(X)$  bits são suficientes, na média, para representar uma sequência de  $n$  variáveis aleatórias independentes e identicamente distribuídas. Neste capítulo, removemos essa restrição para abordar processos estocásticos, nos quais as variáveis aleatórias exibem dependências temporais ou estruturais. Introduziremos a entropia de um processo estocástico, que quantifica a incerteza associada a uma sequência de eventos, e a taxa de entropia, uma medida assintótica da informação gerada por unidade de tempo. Esses conceitos são essenciais para modelar sistemas dinâmicos. Em particular, focaremos nas cadeias de Markov, um caso especial de processos estocásticos em que a dependência é limitada ao estado imediatamente anterior, oferecendo um equilíbrio entre simplicidade e aplicabilidade em problemas de codificação e comunicação.

Um processo estocástico estacionário em sentido estrito é uma sequência de variáveis aleatórias cujas propriedades estatísticas conjuntas permanecem invariantes sob deslocamentos temporais. Isto implica que, por exemplo, médias, variâncias e outras estatísticas não dependem do instante absoluto, permanecem estáveis ao longo do tempo.

**Definição 27** (Processo Estocástico Estacionário (sentido-estrito)).

Uma sequência de v.a.s.  $X_1, X_2, \dots, X_n$  é governada por uma distribuição probabilística é dita estacionária em sentido estrito se

$$p(X_{1:n} = x_{1:n}) = p(X_{1+l:n+l} = x_{1:n}), \quad (245)$$

para todo  $l$ , todo  $n$  e todo  $x_{1:n} \in \mathcal{X}^n$ .

Um processo de Markov é um processo estocástico em que a probabilidade do próximo estado depende apenas do estado atual, e não de toda a história anterior.

**Definição 28** (Processo de Markov de primeira ordem). Um processo estocástico é um processo de Markov de primeira ordem se

$$p(X_{n+1} = x_{n+1} \mid X_{1:n} = x_{1:n}) = p(X_{n+1} = x_{n+1} \mid X_n = x_n) \quad (246)$$

A essência de um processo de Markov reside na ideia de que, dado o estado presente, o futuro e o passado são independentes. Em termos probabilísticos, para o caso de primeira ordem, isto significa que  $p(x_{1:n}) = p(x_1)p(x_2 | x_1) \dots p(x_n | x_{n-1})$ . Essa independência condicional reflete a ‘falta de memória’ do processo, onde conhecer o estado atual basta, não sendo necessário avaliar a influência do passado.

De forma geral, podemos estender a ideia anterior para processos de Markov de ordens superiores.

**Definição 29** (Processo de Markov de ordem  $m$ ). Um processo estocástico é um processo de Markov de ordem  $m$  se

$$p(X_{n+1} = x_{n+1} | X_{1:n} = x_{1:n}) = p(X_{n+1} = x_{n+1} | X_n = x_n, X_{n-1} = x_{n-1}, \dots, X_{n-m} = x_{n-m}). \quad (247)$$

Neste caso, isto significa que  $p(x_{1:n}) = p(x_{m+1} | x_m, x_{m-1}, \dots, p(x_1)) \dots p(x_{n-1} | x_{n-2}, x_{n-3}, \dots, x_{n-m-1})p(x_n | x_{n-1}, x_{n-2}, \dots, x_{n-m})$ .

Uma cadeia de Markov homogênea é um processo de Markov em que as probabilidades de transição entre estados não variam com o tempo.

**Definição 30** (Cadeia de Markov homogênea). Uma cadeia de Markov é invariante no tempo (também chamada de homogênea) se  $p(x_{n+1} | x_n)$  não depender do tempo, i.e., se

$$p(X_{n+1} = b | X_n = a) = p(X_2 = b | X_1 = a) \quad \forall a, b, n. \quad (248)$$

Neste caso, a cadeia de Markov pode ser descrita por uma matriz de transição fixa  $P = [p_{ij}]_{ij}$  em que  $p_{ij} = p(X_{n+1} = j | X_n = i)$ . Podemos representar esta cadeia de Markov como um grafo com setas entre estados cuja probabilidade de transição não é nula. O fato de ser homogênea também facilita o estudo de propriedades assintóticas, como a taxa de entropia, como veremos adiante.

**Exemplo 13** (Cadeia de Markov homogênea de primeira ordem).

Vamos considerar neste exemplo uma cadeia de Markov homogênea de primeira ordem com 4 estados. Esta cadeia pode ser representada pela sua matriz de transição:

$$P = \begin{bmatrix} 0 & p(2|1) & 0 & p(4|1) \\ 0 & p(2|2) & p(3|2) & 0 \\ 0 & 0 & 0 & p(4|3) \\ 0 & 0 & 0 & p(4|4) \end{bmatrix}, \quad (249)$$

ou, alternativamente, pelo grafo ilustrado na Figura 25

Para uma cadeia de Markov homogênea de primeira ordem, a probabilidade de um estado no instante  $n + 1$ , pode ser dada em função

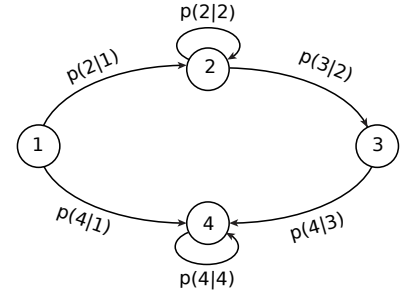


Figura 25: Exemplo de uma cadeia de Markov de primeira ordem com 4 estados.

dos possíveis estados no instante  $n$  e da probabilidade de transição:

$$p(X_{n+1} = i) = \sum_j p(X_n = j)p(i|j), \quad (250)$$

onde  $p(i|j)$  é o elemento  $p_{ij}$  da matriz de transição  $P$ . Esta cadeia de Markov será estacionária se a probabilidade dos estados não mudar ao longo do tempo, ou seja,  $p(X_{n+1} = i) = p(X_n = i)$ , para todo  $i$ . Quando isto ocorrer, haverá uma distribuição estacionária  $\mu$  tal que  $\mu P = \mu$ , onde  $P$  é a matriz de transição. Isto assegura que a probabilidade de cada estado permaneça constante ao longo do tempo quando o processo parte de, ou alcança, uma distribuição  $\mu$ .

**Definição 31** (Cadeira de Markov irredutível). Uma cadeia de Markov é irredutível se  $p_{ij}(n) > 0$  para todo  $i, j$  e para algum  $n$  onde  $p_{ij}(n) = p(X_{n+1} = j \mid X_n = i)$ .

Ou seja, qualquer estado é acessível de qualquer outro estado (ao menos em algum instante  $n$ ), com probabilidade não nula.

**Definição 32** (Período de uma cadeia de Markov). Uma cadeia de Markov é periódica se  $d(i) > 1$  com

$$d(i) = \gcd\{n : p_{ii}(n) > 0\} \quad (251)$$

$d(i)$  é o período do  $i$ -ésimo estado.

Note que temos o máximo divisor comum do número de épocas para o qual um retorno ao mesmo estado é possível.

**Exemplo 14** (Cadeia de Markov homogênea de primeira ordem com dois estados). Suponha uma cadeia de Markov homogênea de primeira ordem com apenas dois estados. Esta cadeia é caracterizada pela matriz de transição

$$P = \begin{pmatrix} 1 - \alpha & \alpha \\ \beta & 1 - \beta \end{pmatrix} \quad (252)$$

e pode também ser representada pelo grafo da Figura 26.

Se  $\mu = [p_1 \ p_2]^T$  é uma distribuição estacionária, então devemos ter  $\mu^T P = \mu^T$ . Neste caso, teremos

$$\mu^T = (p_1 \ p_2) = (p_1 \ p_2) \begin{pmatrix} 1 - \alpha & \alpha \\ \beta & 1 - \beta \end{pmatrix} \quad (253a)$$

$$= ((1 - \alpha)p_1 + \beta p_2 \quad \alpha p_1 + (1 - \beta)p_2) \quad (253b)$$

Obtemos assim:

$$p_1 = (1 - \alpha)p_1 + \beta p_2 \quad (254)$$

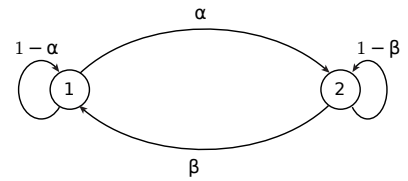


Figura 26: Grafo representando uma cadeia de Markov de primeira ordem com dois estados.

logo,  $p_1 = \frac{\beta}{\alpha} p_2$ . Sabemos também que devemos ter  $p_1 + p_2 = 1$ . Por conseguinte, teremos

$$p_1 + p_2 = 1 \quad (255a)$$

$$\frac{\beta}{\alpha} p_2 + p_2 = 1 \quad (255b)$$

$$p_2 = \frac{\alpha}{\alpha + \beta}. \quad (255c)$$

Assim, teremos

$$\mu = \left( \frac{\frac{\beta}{\alpha + \beta}}{\frac{\alpha}{\alpha + \beta}} \right). \quad (256)$$

A Média de Cesàro representada uma técnica matemática que consiste em substituir a soma parcial tradicional de uma série por uma média aritmética de suas somas parciais. Ela pode ser utilizada para obter representações mais precisas de sinais descontínuos; reduzir artefatos processamento de sinais; aplicada a equações diferenciais parciais; e, em processos estocásticos analisar o comportamento assintótico, especialmente ao estudar convergência ou taxas de entropia.

**Lema 33** (Média Cesáro) *Sejam  $a_n$  números reais, se  $a_n \rightarrow a$  quando  $n \rightarrow \infty$  e  $b_n = \frac{1}{n} \sum_{i=1}^n a_i$ , então  $b_n \rightarrow a$  quando  $n \rightarrow \infty$ .*

*Demonstração.* Como  $a_n \xrightarrow{n \rightarrow \infty} a$ , para todo  $\epsilon > 0$ , existe  $N_\epsilon$  tal que  $|a_n - a| < \epsilon$  para todo  $n > N_\epsilon$ . Para  $n > N_\epsilon$  teremos

$$|b_n - a| = \left| \frac{1}{n} \sum_{i=1}^n a_i - a \right| = \left| \frac{1}{n} \sum_{i=1}^n a_i - \frac{1}{n} \sum_{i=1}^n a \right| \quad (257a)$$

$$= \left| \frac{1}{n} \sum_{i=1}^n (a_i - a) \right| \leq \frac{1}{n} \sum_{i=1}^n |a_i - a| \quad (257b)$$

$$= \frac{1}{n} \left( \sum_{i=1}^{N_\epsilon} |a_i - a| + \sum_{i=N_\epsilon+1}^n |a_i - a| \right) \quad (257c)$$

$$\leq \frac{1}{n} \sum_{i=1}^{N_\epsilon} |a_i - a| + \frac{1}{n} \sum_{i=N_\epsilon+1}^n \epsilon \quad (257d)$$

$$= \frac{1}{n} \sum_{i=1}^{N_\epsilon} |a_i - a| + \underbrace{\frac{n - N_\epsilon}{n}}_{< 1} \epsilon \quad (257e)$$

$$< \underbrace{\frac{1}{n} \sum_{i=1}^{N_\epsilon} |a_i - a|}_{< \epsilon} + \epsilon < 2\epsilon \quad (257f)$$

onde em 257b utilizamos a desigualdade triangular e em 257f utilizamos o fato de que podemos tomar  $n$  grande suficiente de forma que  $\frac{1}{n} \sum_{i=1}^{N_\epsilon} |a_i - a| < \epsilon$ , pois trata-se de uma soma finita.

Então  $b_n \rightarrow a$  quando  $n \rightarrow \infty$ .

□



Processos Estocástico possuem taxas de entropia, que intuitivamente representam o quantidade de informação nova, na média, que é fornecida pelo processo estocástico a cada instante.

**Definição 33** (Taxa de Entropia de um processo estocástico). A taxa de entropia de um processo estocástico  $\{X_i\}_i$  é definida como

$$H(\mathcal{X}) \triangleq \lim_{n \rightarrow \infty} \frac{1}{n} H(X_1, X_2, \dots, X_n), \quad (258)$$

quando existir.

Observe que, na Equação (258), quando as v.a.s são i.i.d. teremos  $H(X_1, \dots, X_n) = H(X_1) + \dots + H(X_n) = nH(X)$  e assim  $H(\mathcal{X}) = H(X)$ , ou seja,

$$H(\mathcal{X}) = \lim_{n \rightarrow \infty} \frac{1}{n} H(X_{1:n}) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n H(X_i) = H(X_1). \quad (259)$$

A taxa de entropia pode ser vista como a entropia por símbolo para um dado processo estocástico, quando  $n$  cresce indefinidamente.

A Taxa de Entropia de um processo estocástico  $H(\mathcal{X})$  mede a incerteza média por símbolo em uma sequência longa, enquanto a Taxa de Inovação da Informação  $H'(\mathcal{X})$ , que será definida a seguir, quantifica a incerteza de um novo símbolo dado o passado. Para processos gerais, essas taxas podem diferir devido à dependência entre símbolos; no entanto, em processos estacionários, ambas convergem ao mesmo valor, ou seja,  $H(\mathcal{X}) = H'(\mathcal{X})$ , refletindo a estabilidade estatística do processo ao longo do tempo.

**Definição 34** (Taxa de Inovação da Informação). Vamos assumir um processo estocástico e definir a taxa da seguinte forma

$$H'(\mathcal{X}) \triangleq \lim_{n \rightarrow \infty} H(X_n \mid X_{n-1}, X_{n-2}, \dots, X_1), \quad (260)$$

se existir.

Veremos a seguir que  $H'(\mathcal{X})$  existe para um processo estocástico estacionário.

**Teorema 34** (Um processo estocástico estacionário possui taxa de inovação de informação) *Para um processo estocástico estacionário,  $H(X_n \mid X_{n-1}, X_{n-2}, \dots, X_1)$  é decrescente com  $n$  e possui como limite  $H'(\mathcal{X})$ .*

*Demonstração.*

$$H(X_{n+1} \mid X_1, \dots, X_n) \leq H(X_{n+1} \mid X_2, \dots, X_n) \quad (261a)$$

$$= H(X_n \mid X_1, \dots, X_{n-1}) \quad (261b)$$

Onde utilizamos o fato de que condicionar não aumenta (decrece ou não altera) a entropia; e utilizamos o fato de que o processo estocástico é estacionário.

Temos então uma sequência decrescente com limite inferior 0, logo, esta sequência possui um limite:  $H'(\mathcal{X})$ .

□

**Teorema 35** (Em um processo estocástico estacionário taxa de inovação é igual a taxa de entropia) *Para um processo estocástico estacionário temos*

$$\lim_{n \rightarrow \infty} H(X_n | X_{n-1}, X_{n-2}, \dots, X_1) \triangleq H'(\mathcal{X}) \quad (262a)$$

$$= H(\mathcal{X}) \triangleq \lim_{n \rightarrow \infty} \frac{1}{n} H(X_1, X_2, \dots, X_n) \quad (262b)$$

*Demonstração.*

$$b_n = \frac{H(X_1, X_2, \dots, X_n)}{n} = \frac{1}{n} \sum_{i=1}^n \underbrace{H(X_i | X_{i-1}, \dots, X_1)}_{=a_i}, \quad (263)$$

como  $a_n \rightarrow H'(\mathcal{X})$ , teremos  $b_n \rightarrow H'(\mathcal{X})$ , mas por definição  $b_n \rightarrow H(\mathcal{X})$ .

□

**Exemplo 15** (Taxa de entropia para cadeia de Markov de primeira ordem). A taxa de entropia para uma cadeia de Markov de primeira ordem estacionária será dada da seguinte forma

$$H(\mathcal{X}) = H'(\mathcal{X}) = \lim_{n \rightarrow \infty} H(X_n | X_{n-1}, \dots, X_1) \quad (264a)$$

$$= \lim_{n \rightarrow \infty} H(X_n | X_{n-1}) \quad (264b)$$

$$= H(X_2 | X_1) \quad (264c)$$

$$= - \sum_{x_2, x_1} p(x_2, x_1) \log p(x_2 | x_1) \quad (264d)$$

$$= \sum_i \mu_i \left[ - \sum_j p_{ij} \log p_{ij} \right], \quad (264e)$$

onde  $\mu$  é a distribuição estacionária,  $p_{ij}$  a probabilidade de transição de  $i$  para  $j$  e, em 264c, utilizamos o fato de termos uma cadeia de Markov de primeira ordem estacionária.

**Exemplo 16** (Markov com apenas dois estados). Para a cadeia de Markov simples apresentada no Exemplo 14, teremos

$$H(\mathcal{X}) = H(X_2 | X_1) = \frac{\beta}{\alpha + \beta} H(\alpha) + \frac{\alpha}{\alpha + \beta} H(\beta). \quad (265)$$

**Exemplo 17** (Modelos de Ehrenfest). Os Modelos de Ehrenfest são cadeias de Markov que descrevem a dinâmica de partículas em um sistema com dois compartimentos. Tais modelos são frequentemente usados para ilustrar conceitos de equilíbrio estatístico e comportamento assintótico. A análise desse processo estocástico revela propriedades como estacionaridade e convergência para uma distribuição de equilíbrio. Paul Ehrenfest propôs um modelo simples para a troca de calor ou moléculas entre dois corpos isolados. Neste modelo da cadeia de Ehrenfest iremos considerar a troca de moléculas de gás entre dois compartimentos, rotulados por 0 e 1, conforme apresentado na Figura 27.

Partículas movem-se aleatoriamente entre os compartimentos com probabilidades fixas, e o estado do sistema é representado pelo número de partículas em um dos compartimentos. Iremos denotar por  $X_k$  o número de partículas no compartimento 1 no instante  $k$ .  $X_k$  descreve o estado atual do sistema. Temos um processo estocástico gerando  $X_{1:n}$ , onde  $X_k \in \{0, 1, \dots, m\}$ .

Supondo que no instante  $k$  o compartimento 1 possua  $i$  partículas ( $X_k = i$ ). O modelo de Ehrenfest nos fornece:

$$\Pr(X_{k+1} = i + 1 | X_k = i) = \frac{m - i}{m}, \quad (266a)$$

$$\Pr(X_{k+1} = i - 1 | X_k = i) = \frac{i}{m}. \quad (266b)$$

Neste modelo a propriedade de Markov é satisfeita:

$$\Pr(X_{n+1} = i + 1 | X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}, X_n = i) = \Pr(X_{n+1} = i + 1 | X_n = i), \quad (267a)$$

$$\Pr(X_{n+1} = i - 1 | X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}, X_n = i) = \Pr(X_{n+1} = i - 1 | X_n = i). \quad (267b)$$

A distribuição condicional não depende de  $n$ , desta forma, temos um processo estocástico homogêneo. A cadeia de Markov poderá então ser descrita por uma matriz de transição fixa  $P = [p_{ij}]_{ij}$ .

Para  $m = 4$  teremos a cadeia de Markov representada pelo grafo apresentado na Figura 28.

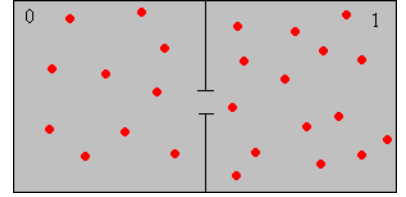


Figura 27: Exemplo de uma cadeia de Markov de primeira ordem com 4 estados.

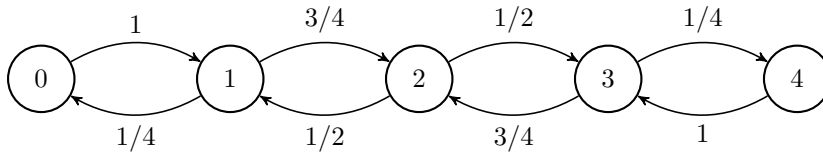


Figura 28: Modelo de Ehrenfest com 4 estados.

Para este modelo, a matriz de transição é

$$\mathbf{P} = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & 4 \end{matrix} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1/4 & 0 & 3/4 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 3/4 & 0 & 1/4 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} & \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{matrix} \end{matrix} \quad (268)$$

A distribuição de estado estacionário é tal que  $\mu^T P = \mu^T$ , com  $\sum_i \mu_i = 1$ .

$$\mu^T P = \mu^T \quad (269a)$$

$$\mu^T (P - I) = 0 \quad (269b)$$

$$\mu^T Q = 0 \quad (269c)$$

Teremos aqui

$$Q = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 1/4 & -1 & 3/4 & 0 & 0 \\ 0 & 1/2 & -1 & 1/2 & 0 \\ 0 & 0 & 3/4 & -1 & 1/4 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}. \quad (270)$$

Vamos incorporar a condição  $\sum_i \mu_i = 1$  fazendo:

$$\tilde{Q} = \begin{pmatrix} -1 & 1 & 0 & 0 & 1 \\ 1/4 & -1 & 3/4 & 0 & 1 \\ 0 & 1/2 & -1 & 1/2 & 1 \\ 0 & 0 & 3/4 & -1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad (271)$$

e o novo sistema de equações será

$$\mu^T \tilde{Q} = (0, 0, 0, 1). \quad (272)$$

Para solucionar a Equação 272, iremos pós multiplicar ambos os lados pela matriz inversa de  $\tilde{Q}$ ,

$$\mu^T = (0, 0, 0, 1) \tilde{Q}^{-1}, \quad (273)$$

obtendo assim

$$\mu^T = \left( \frac{1}{16}, \frac{1}{4}, \frac{6}{16}, \frac{1}{4}, \frac{1}{16} \right) \quad (274)$$

Conforme vimos na Equação (264), a taxa de entropia para um cadeia de Markov de primeira ordem estacionária será dada da seguinte forma

$$H(\mathcal{X}) = \sum_i \mu_i H(\mathbf{p}_i), \quad (275)$$

onde  $\mu$  é a distribuição estacionária,  $p_{ij}$  a probabilidade de transição de  $i$  para  $j$  (elementos da matriz  $P$ ) e  $\mathbf{p}_i$  a  $i$ -ésima linha da matriz  $P$  (as probabilidades de transição à partir do estado  $i$ ).

Para o exemplo em questão:

$$H(\mathcal{X}) = \mu_1 H(\mathbf{p}_1) + \mu_2 H(\mathbf{p}_2) + \mu_3 H(\mathbf{p}_3) + \mu_4 H(\mathbf{p}_4) + \mu_5 H(\mathbf{p}_5) \quad (276a)$$

$$= \frac{1}{16} H(0, 1, 0, 0, 0) + \frac{1}{4} H\left(\frac{1}{4}, 0, \frac{3}{4}, 0, 0\right) + \frac{6}{16} H\left(0, \frac{1}{2}, 0, \frac{1}{2}, 0\right) + \frac{1}{4} H\left(0, 0, \frac{3}{4}, 0, \frac{1}{4}\right) + \frac{1}{16} H(0, 0, 0, 1, 0) \quad (276b)$$

$$= 0 + 2 \times \frac{1}{4} \left( \frac{1}{4} \log 4 + \frac{3}{4} \log \frac{4}{3} \right) + \frac{6}{16} + 0 \quad (276c)$$

$$= \frac{1}{2} \left( \frac{1}{2} + \frac{3}{2} - \frac{3}{4} \log 3 \right) + \frac{6}{16} \quad (276d)$$

$$= 1 - \frac{3}{8} \log 3 + \frac{6}{16} = \frac{11}{8} - \frac{3}{8} \log 3 = 0.78064 \quad (276e)$$

# Codificação de Fonte

A codificação de fonte sem perdas é o objetivo central quando buscamos representar os símbolos de uma fonte estocástica de forma eficiente. Ou seja, desejamos utilizar o menor número de bits possível para codificar a informação produzida pela fonte, sem que haja perda de informação, sendo assim possível reconstruir no receptor, exatamente a mesma mensagem enviada pelo emissor.

O Teorema da Codificação de Shannon estabelece que, para uma fonte com entropia  $H(X)$ , é possível codificar utilizando uma taxa  $R > H(X)$  bits por símbolo, desde que se empregue blocos suficientemente grandes. No entanto, a codificação por blocos pode ser impraticável em aplicações reais devido à latência e à complexidade computacional, o que motiva a busca por alternativas mais práticas que ainda alcancem o limite teórico da entropia.

Neste capítulo, exploraremos duas abordagens principais: códigos de tamanho variável, como a Codificação Huffman, que atribui palavras-código de comprimento variável a símbolos individuais, e a codificação de fluxo (*stream coding*), que opera no fluxo de dados considerando o símbolo atual e sua história, como na Codificação Aritmética e na Codificação Lempel-Ziv — esta última sendo universal, pois não requer conhecimento prévio da distribuição  $p(x)$ . Assumiremos que a distribuição  $p(x)$  da fonte é conhecida ou que uma aproximação  $q(x)$  está disponível, e investigaremos o impacto de utilizar  $q(x)$  quando a distribuição verdadeira é  $p(x)$ , sem abordar diretamente o problema de estimação de  $p(x)$ . Buscaremos analisar métodos práticos que se aproximem do limite de entropia.

Para abordar este assunto, devemos inicialmente fazer algumas definições.

Um código, no contexto de codificação de fonte, é uma função que mapeia os símbolos de uma fonte em sequências de bits (palavras-código) de forma a representá-los. Usualmente, o objetivo em se estabelecer um código é minimizar o número de bits necessário para transmissão ou armazenamento da mensagem produzida pela fonte. Um código pode ser de tamanho fixo ou variável, e, para codificação sem perdas, devemos requerer que cada símbolo seja univocamente

recuperado a partir de sua palavra-código.

**Definição 35** (Codificação de Fonte). Um código  $C$  para a v.a.  $X$  é um mapeamento de  $\mathcal{X}$  em  $\mathcal{D}^*$ , ou seja

$$C : \mathcal{X} \rightarrow \mathcal{D}^*, \quad (277)$$

onde  $\mathcal{D}^*$  é o conjunto de seqüências (*strings*) finitas em um alfabeto  $D$ -ário.  $C(x)$  é a palavra código (*codeword*) correspondente ao símbolo  $x$ , e  $l(x)$  é o comprimento desta palavra.

De forma geral,  $\mathcal{D} = \{0, 1, \dots, D-1\}$ , mas usualmente utilizamos  $D = 2$ , ou seja, codificação binária com alfabeto  $\mathcal{D} = \{0, 1\}$ . O conjunto  $\mathcal{D}^*$  representa a extensão do alfabeto  $\mathcal{D}$  (fecho de Kleene<sup>14</sup> do alfabeto), consistindo de todas as seqüências finitas (de qualquer tamanho, incluindo a seqüência vazia) formadas por símbolos de  $\mathcal{D}$ , excluindo seqüências infinitas, que seriam denotadas por  $\mathcal{D}^\infty$ .

<sup>14</sup> O fecho de Kleene de um alfabeto  $\mathcal{D}$ , denotado  $\mathcal{D}^*$ , é um conceito da teoria de linguagens formais que inclui todas as seqüências finitas que podem ser formadas concatenando zero ou mais símbolos de  $\mathcal{D}$ , incluindo a seqüência vazia, frequentemente representada por  $\epsilon$ .

**Exemplo 18.** Seja  $\mathcal{X} = \{\text{azul}, \text{vermelho}\}$ . O código pode ser  $C(\text{vermelho}) = 00$  e  $C(\text{azul}) = 11$ , o que seria um código binário para  $\mathcal{D} = \{0, 1\}$ . Teríamos então  $\mathcal{D}^* = \{00, 11\}$ . Uma seqüência produzida pela fonte da forma (azul, azul, vermelho) seria codificada como (11, 11, 00), ou melhor, 111100.

O comprimento esperado de um código é uma medida fundamental na codificação de fonte, pois quantifica a eficiência média do código ao representar os símbolos da fonte, permitindo comparar diferentes esquemas de codificação em termos de uso de bits.

**Definição 36** (Comprimento Esperado). O comprimento esperado  $L(C)$  de um código  $C$  para uma v.a.  $X$  com distribuição  $p(x)$  é dado por

$$L(C) = \sum_x p(x)l(x). \quad (278)$$

**Exemplo 19.** Seja  $\mathcal{X} = \{1, 2, 3, 4\}$  e  $\mathcal{D} = \{0, 1\}$ . Podemos definir o código através da Tabela 1. Neste caso, teremos  $H(X) = 1.75$  e

x	p(x)	c(x)	l(x)
1	1/2	0	1
2	1/4	10	2
3	1/8	110	3
4	1/8	111	3

Tabela 1: Código binário de exemplo para quatro símbolos.

$L(C) = El(X) = 1.75$ , então este código é muito bom. A decodificação para este código é fácil, por exemplo, dada a seqüência binária 01011011111110100, podemos prontamente identificar a seqüência de símbolos que a produziu: 1,2,3,4,4,3,2,1. Seria como se tivéssemos uma pontuação separando os símbolos: 0,10,110,111,111,110,10,0. Dizemos que o código possui pontuação automática.

**Exemplo 20.** Vamos considerar  $\mathcal{X} = \{1, 2, 3\}$  e  $\mathcal{D} = \{0, 1\}$ . O código proposto é apresentado na Tabela 2. Para o código proposto,

x	1	2	3
$p(x)$	1/3	1/3	1/3
$C(X)$	0	10	11

Tabela 2: Exemplo de código binário para um alfabeto com 3 símbolos.

teremos então  $El(X) = 1.66 > H$  bits, uma vez que  $H = 1.58$  bits. Note ainda como podemos facilmente decodificar uma sequência, como  $10110010 = 2, 3, 1, 1, 2$ .

A eficiência de um código é uma medida que compara o comprimento esperado  $L$  do código com a entropia  $H(X)$  da fonte. Ela serve como um indicativo do quão próximo o código está do limite teórico de compressão estabelecido pelo Teorema da Codificação de Shannon.

**Definição 37** (Eficiência de um código). A eficiência de um código é definida da seguinte forma

$$0 \leq \text{eficiência} \triangleq \frac{H_D(X)}{El(X)} \leq 1. \quad (279)$$

**Exemplo 21.** Para  $\mathcal{X} = \{1, 2, 3, 4\}$  e  $\mathcal{D} = \{0, 1\}$  (código binário), considere os seguintes códigos dados na Tabela 3. Neste caso, obtemos

$x$	$p(x)$	$C_I$	$C_{II}$	$C_{III}$	$C_{IV}$
1	0.5	0	0	0	0
2	0.25	0	1	10	01
3	0.125	1	00	110	011
4	0.125	10	11	111	0111
$H(X)$	1.75	-	-	-	-
$El(X)$	-	1.125	1.25	1.75	1.875
eficiência		1.555	1.4	1	0.933

Tabela 3: Comparativo de 4 códigos distintos para uma fonte com 4 símbolos e distribuição dada.

valores impróprios para os códigos  $C_I$  e  $C_{II}$ , indicando que não podem ser códigos sem perdas, já que violam do teorema de Shannon. O valor esperado do comprimento do código não pode ser menor que a entropia sem que o código incorra em erro.

Note que o código  $C_I$  não é unívoco. A sequência 00001 pode ser decodificada como 1, 2, 1, 2, 3 ou 2, 1, 2, 1, 3 ou 1, 1, 1, 1, 3.

Considere  $C_{II}$ . O resultado da decodificação de 0011 poderia ser 1, 1, 2, 2 ou 3, 4 ou 3, 2, 2 ou 1, 1, 4. Novamente, não é possível decodificar sem que a probabilidade de erro seja nula (quando a sequência torna-se mais longa, a probabilidade de erro tende a 1).

O código  $C_{III}$  é factível (já que  $El \geq H$ ). O processo de decodificação é simples. Por exemplo:  $111110100 = 111, 110, 10, 0 = 4, 3, 2, 1$ .

Utilizando o código  $C_{IV}$  é possível decodificar a sequência 00000000111 univocamente, porém os símbolos não são instantaneamente decodificados, é necessário analisar os bits à frente para conseguir decodificar.



**Definição 38** (Extensão de um código). Um extensão  $C^*$  de um código  $C$  é um mapeamento de uma sequência finita de símbolos em  $\mathcal{X}$  em uma sequência finita de símbolos em  $\mathcal{D}$  através da concatenação dos códigos:

$$C^*(x_1, x_2, \dots, x_n) = C(x_1)C(x_2) \dots C(x_n). \quad (280)$$

Um código não-singular é uma função de codificação que mapeia os símbolos de uma fonte estocástica em palavras-código de forma que cada símbolo tenha uma representação única, sendo essa uma condição necessária para a codificação sem perdas, embora não garanta por si só a decodificação unívoca.

**Definição 39** (Código não-singular). Um código é dito **não-singular** se todo elemento de  $\mathcal{X}$  é mapeado em sequências (*strings*) diferentes em  $\mathcal{D}^*$ . I.e.,

$$x_i \neq x_j \Rightarrow C(x_i) \neq C(x_j). \quad (281)$$

Para garantir que cada sequência codificada possa ser recuperada de forma inequívoca a partir de sua representação em bits, um código precisa ter decodificação unívoca, ou seja, não pode haver ambiguidades na decodificação.

**Definição 40** (Código com decodificação unívoca). Um código  $C$  com extensão  $C^*$  é decodificável univocamente se a sua extensão  $C^*$  for não-singular.

Em um código de prefixo nenhuma palavra-código é um prefixo de outra, garantindo que a decodificação possa ser realizada de forma unívoca e sequencial sem a necessidade de olhar para além do próximo delimitador.

**Definição 41** (Código de prefixo). Um código é chamado *código de prefixo* ou *código instantâneo* se nenhuma palavra é prefixo de qualquer outra palavra.

Quando temos um código de prefixo, sabemos onde está o fim de uma palavra, pois ela não é prefixo de nenhuma outra, ou seja, não existe nenhuma outra palavra que comece com a palavra dada. Podemos dizer que um código de prefixo possui então a propriedade de auto-pontuação.

Dentre o conjunto dos códigos unívocos possíveis, queremos um código com comprimento esperado mínimo possível. Analisando as classes de códigos, intuitivamente pensamos que é mais provável encontrarmos um código com comprimento esperado menor em uma classe maior (ou mais abrangente). A Figura 29 apresentado a estrutura hierárquica dos códigos (todos os códigos  $\supset$  não-singulares  $\supset$  unicamente decodificáveis  $\supset$  prefixo).



Figura 29: Diagrama de representação dos conjuntos de códigos, evidenciando a estrutura hierárquica existente.

### Desigualdade de Kraft

A Desigualdade de Kraft estabelece uma condição necessária e suficiente para a existência de códigos de prefixo com comprimentos de palavras-código especificados. A demonstração, que será vista, ainda fornecerá um método para construir um código de prefixo.

**Teorema 36** (Desigualdade de Kraft) *Para qualquer código instantâneo (código de prefixo) sobre um alfabeto de tamanho  $D$ , o comprimento das palavras  $l_1, l_2, \dots, l_m$  deve satisfazer*

$$\sum_i D^{-l_i} \leq 1. \quad (282)$$

*Por um outro lado, dado um conjunto de comprimentos de código satisfazendo a desigualdade acima, então existe um código de prefixo com estes comprimentos.*

Note que foi dito que existe um código de prefixo com aqueles comprimentos, não significa que todos códigos cujos comprimentos satisfazem a desigualdade são códigos de prefixo. Se existe um código não-instantâneo com comprimentos  $l_i$  satisfazendo a desigualdade de Kraft, então podemos encontrar um outro código, que terá a propriedade de prefixo, e que possuirá os mesmos comprimentos  $l_i$ , consequentemente não alterando o comprimento esperado. Logo, será sempre melhor escolher um código de prefixo.

*Demonstração (Desigualdade de Kraft).* Vamos representar o conjunto de códigos em uma árvore  $D$ -ária (não necessariamente balanceada), conforme a Figura 30.

As palavras correspondem às folhas na árvore. O caminho da raiz até a folha determina a palavra código. A condição de prefixo implica que não existe uma palavra a não ser nas folhas (nenhum descendente de uma palavra código será também uma palavra código).  $l_{\max} = \max_i(l_i)$  é o comprimento da palavra mais longa. Podemos expandir toda a árvore até o comprimento  $l_{\max}$ . Os nós no nível de  $l_{\max}$  são: palavras de código; descendentes de palavras de código; ou apenas nós sem palavra código associada a ele ou seus antecessores. Considere uma palavra  $i$  no nível  $l_i$  da árvore (então esta palavra possui comprimento  $l_i$ ). Existem então  $D^{l_{\max}-l_i}$  descendentes de  $i$ , na árvore, no nível  $l_{\max}$ . Com a condição de prefixo, podemos afirmar que os descendentes de código  $i$  no nível  $l_i$  são disjuntos dos descendentes do código  $j$  no nível  $l_j$ , quando  $i \neq j$  (i.e., o conjunto de descendentes para diferentes palavras é disjunto). O número total de nós em um conjunto de todos descendentes é  $\leq D^{l_{\max}}$ . Utilizando o que vimos anteriormente, temos que a soma do número de descendentes de todos

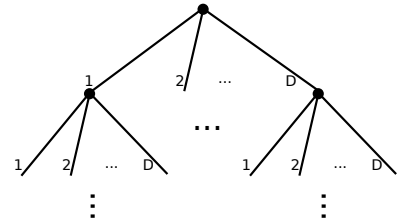


Figura 30: Árvore  $D$ -ária.

os códigos é menor ou igual ao número de folhas na árvore cheia no nível  $l_{\max}$ . Podemos então escrever

$$\sum_i D^{l_{\max}-l_i} \leq D^{l_{\max}} \Rightarrow \sum_i D^{-l_i} \leq 1. \quad (283)$$

Por outro lado, dados os comprimentos  $l_1, l_2, \dots, l_m$ , satisfazendo Kraft, iremos mostrar como construir um código de prefixo utilizando estes comprimentos. Considere uma árvore  $D$ -ária cheia de profundidade  $l_{\max}$  e  $D^{l_{\max}}$  nós terminais. Observe que: no nível 0 existe uma fração 1 de descendentes de cada nó neste nível; no nível 1 existe uma fração  $1/D$  de descendentes de cada nó neste nível; no nível 2 existe uma fração  $1/D^2$  de descendentes de cada nó neste nível; e assim por diante. De forma geral, em cada nível  $i \in [0, l_{\max}]$  da árvore, existe uma fração de  $D^{-i}$  nós terminais que são descendentes de uma ramificação de cada um dos  $D^i$  nós no nível  $i$ . Ordene então os comprimentos  $(l_1, l_2, \dots, l_m)$  de forma ascendente  $(s_1, s_2, \dots, s_m)$ , sendo  $s_1 \leq s_2 \leq \dots \leq s_m$ . Observe que existem tantos comprimentos quanto existem palavras. Para o comprimento  $s_1$ , escolha um nó no nível  $s_1$  para indicar o código. Para garantir a condição de prefixo, o nó escolhido deve se tornar um nó terminal, eliminando assim uma fração  $D^{-s_1}$  de nós terminais no nível  $l_{\max}$ . Em seguida, escolha um dos nós remanescentes no nível  $s_2$  (neste momento, existem  $(D^{s_1} - 1)D^{s_2-s_1}$  possíveis escolhas), eliminando assim uma fração  $D^{-s_2}$  de nós terminais no nível  $l_{\max}$ . A fração total de nós eliminados, até o momento, foi de  $D^{-s_1} + D^{-s_2}$ . Continuando este processo, iremos eliminar uma fração  $\sum_{i=1}^m D^{-s_i}$  de nós, e neste processo estamos garantindo que estamos criando um código instantâneo (uma palavra de código não pode ser prefixo de outra). Como por suposição temos  $\sum_{i=1}^m D^{-s_i} \leq 1$ , nunca iremos eliminar mais do que todas as palavras de código, então este processo não irá exaurir as palavras de código. Criamos assim um código de prefixo com os comprimentos desejados, satisfazendo Kraft.

□

A desigualdade de Kraft é uma condição *sine qua non* (necessária e indispensável) para a existência de um código prefixo. A condição dada na Equação (282) deve ser satisfeita para que um dado código com comprimentos  $l_1, l_2, \dots, l_m$  seja um código de prefixo. A demonstração também nos mostra como construir um código de prefixo, dado um conjunto de comprimentos  $\{l_1, l_2, \dots, l_m\}$  que satisfaça a Equação (282).

### *Em busca do código ótimo*

Desejamos um código que forneça decodificação instantânea e livre de ambiguidade, além disso, queremos um código eficiente, que tenha

comprimento esperado mínimo. Desta forma, estamos em busca de um código de prefixo ótimo.

O comprimento esperado é dado por

$$L(C) = \sum_i p_i l_i, \quad (284)$$

onde os comprimentos  $l_i$ 's devem ser números inteiros. Teremos então o seguinte problema de otimização com restrição:

$$\min_{\{l_{1:m}\} \in \mathbb{Z}_+^m} L(C) = \sum_i p_i l_i, \quad (285)$$

sujeito a

$$\sum_i D^{-l_i} \leq 1. \quad (286)$$

Este é um problema de programação em inteiros que é um problema cuja sua complexidade computacional cresce exponencialmente com o tamanho da entrada. Não seria assim possível resolvê-lo de forma eficiente. Vamos relaxar a condição de que  $l_i$  precisa ser inteiro e considerar o Lagrangiano

$$J = \sum_i p_i l_i + \lambda \left( \sum_i D^{-l_i} - 1 \right) \quad (287)$$

Para encontrar os pontos estacionários de  $J$ , considerado uma função dos  $l_i$ 's e do multiplicador lagrangiano  $\lambda$ , devemos encontrar as condições para que as derivadas se igualam a zero, ou seja,  $\partial J / \partial l_i = 0$ ,  $i = 1, \dots, m$ , e  $\partial J / \partial \lambda = 0$ . Teremos assim

$$\frac{\partial J}{\partial l_i} = p_i - \lambda D^{-l_i} \ln D = 0, \quad (288)$$

e assim

$$D^{-l_i} = \frac{p_i}{\lambda \ln D}. \quad (289)$$

Teremos também

$$\frac{\partial J}{\partial \lambda} = \sum_i D^{-l_i} - 1 = 0, \quad (290)$$

que nos fornecerá

$$\lambda = 1 / \ln D. \quad (291)$$

Utilizando as Equações (289) e (291) obtemos  $D^{-l_i} = p_i$ , e assim os comprimentos ótimos são dados por

$$l_i^* = -\log_D p_i. \quad (292)$$

Desta forma, o comprimento esperado do código ótimo será

$$L^* = \sum_i p_i l_i^* = -\sum_i p_i \log_D p_i = H_D(X) = H(X) / \log D. \quad (293)$$

Assumindo que seja possível utilizar comprimentos fracionários, os comprimentos ótimos para as palavras de um código são  $l_i^* = -\log_D p_i$ , isto implica que os comprimentos ótimos são iguais à informação do evento (o menor comprimento para codificação de um evento é inerentemente a informação sobre o próprio evento — princípio da Navalha de Occam<sup>15</sup>), e assim o comprimento esperado do código será a entropia.

Apesar os comprimentos ótimos dados na Equação (292) não seja, a princípio, inteiros, podemos utilizar a codificação em blocos e assim aproximar do limite ótimo mesmo quando os comprimentos dos códigos por símbolos são valores fracionários.

<sup>15</sup> A Navalha de Occam, princípio atribuído ao frade franciscano Guilherme de Ockham (século XIV), sugere que, entre explicações concorrentes, a mais simples — aquela que requer menos suposições — deve ser preferida, desde que explique os fatos adequadamente.

**Teorema 37** (Comprimento esperado mínimo) *Entropia é o comprimento esperado mínimo. O comprimento esperado  $L$  de qualquer código  $D$ -ário instantâneo (satisfaz Kraft por conseguinte) para uma v.a.  $X$  é tal que*

$$L \geq H_D(X), \quad (294)$$

com igualdade sse  $D^{-l_i} = p_i$ .

*Demonstração.*

$$L - H_D(X) = \sum_i p_i l_i - \sum_i p_i \log_D 1/p_i \quad (295a)$$

$$= - \sum_i p_i \log_D D^{-l_i} + \sum_i p_i \log_D p_i \quad (295b)$$

$$= - \sum_i p_i \log_D D^{-l_i} + \underbrace{\log_D \left( \sum_i D^{-l_i} \right) - \log_D \left( \sum_i D^{-l_i} \right)}_{=0} + \sum_i p_i \log_D p_i \quad (295c)$$

$$= \sum_i p_i \log_D \frac{p_i}{r_i} - \log_D \left( \sum_i D^{-l_i} \right) \quad (295d)$$

$$= \underbrace{D(p \parallel r)}_{\geq 0} + \underbrace{\log_D(1/c)}_{\geq 0} \quad (295e)$$

$$\geq 0 \quad (295f)$$

onde utilizamos em 295c, para simplificar, a soma de três das quatro parcelas é dada na Equação (296), e em 295f utilizamos que  $c \leq 1$  pois

a desigualdade de Kraft deve ser satisfeita, onde  $c = \sum_i D^{-l_i}$ .

$$-\sum_i p_i \log_D D^{-l_i} + \log_D \left( \sum_j D^{-l_j} \right) + \sum_i p_i \log_D p_i =$$

(296a)

$$-\sum_i p_i \log_D D^{-l_i} + \left( \underbrace{\sum_i p_i}_{=1} \right) \log_D \left( \sum_i D^{-l_i} \right) + \sum_i p_i \log_D p_i$$

(296b)

$$-\sum_i p_i \log_D D^{-l_i} + \sum_i p_i \log_D \left( \sum_j D^{-l_j} \right) + \sum_i p_i \log_D p_i =$$

(296c)

$$\sum_i p_i \log_D \frac{p_i \left( \sum_j D^{-l_j} \right)}{D^{-l_i}} =$$

(296d)

$$\sum_i p_i \log_D \frac{p_i}{\frac{D^{-l_i}}{\left( \sum_j D^{-l_j} \right)}} = \sum_i p_i \log_D \frac{p_i}{r_i},$$

(296e)

onde definimos

$$r_i = \frac{D^{-l_i}}{\sum_j D^{-l_j}}.$$

(297)

□

Assumindo que Kraft seja satisfeito (existe um código de prefixo com tais comprimentos) teremos comprimentos inteiros de forma que  $El \geq H$ . Se ainda  $l_i = -\log_D p_i$  é inteiro, para todo  $i$ , então teremos a igualdade  $El = H$ . Se  $l_i \neq -\log_D p_i$ , mas  $l_i$  é inteiro, teremos estritamente  $El > H$ .

### *Código de Shannon*

O código de Shannon é um código subótimo que consiste em considerar o comprimento como inteiro mais próximo do valor ótimo, ou seja,

$$l_i = \lceil \log_D 1/p_i \rceil.$$

(298)

Os comprimentos definidos desta forma satisfazem Kraft, uma vez que

$$\sum_i D^{-l_i} = \sum_i D^{-\lceil \log_D 1/p_i \rceil} \leq \sum_i D^{-\log_D 1/p_i} = \sum_i p_i = 1. \quad (299)$$

Pelo Teorema 36, existe então um código de prefixo com os comprimentos dados em 298.

**Teorema 38** (Limites para o comprimento esperado do código de Shannon) *Sejam  $l_1, l_2, \dots, l_m$  comprimentos inteiros do código de Shannon para uma fonte  $p$  e um alfabeto  $D$ -ário.  $L$  é o comprimento esperado. Então*

$$H_D(X) \leq L \leq H_D(X) + 1. \quad (300)$$

*Demonstração.* Para o código de Shannon, teremos os seguintes limites:

$$\log_D \frac{1}{p_i} \leq l_i \leq \log_D \frac{1}{p_i} + 1 \quad (301a)$$

$$E_p \left[ \log_D \frac{1}{p_i} \right] \leq E_p [l_i] \leq E_p \left[ \log_D \frac{1}{p_i} + 1 \right] \quad (301b)$$

$$\sum_i p_i \log_D \frac{1}{p_i} \leq L \leq \sum_i p_i \left( \log_D \frac{1}{p_i} + 1 \right) \quad (301c)$$

$$H_D(X) \leq L \leq H_D(X) + 1. \quad (301d)$$

□

Observe ainda que,  $H \leq L^* \leq L$ , onde  $L^*$  é o comprimento ótimo para códigos de comprimento inteiro. Desta forma, também teremos a seguinte relação para o comprimento esperado de código ótimo:  $H_D(X) \leq L^* \leq H_D(X) + 1$ .

Podemos melhorar a eficiência do código de Shannon codificando mais de um símbolo por vez (codificação em bloco). Vamos chamar de  $L_n$  o comprimento esperado por símbolo de uma sequência de  $n$  símbolos  $x_{1:n}$ . Então,

$$L_n = \frac{1}{n} \sum_{x_{1:n} \in \mathcal{X}^n} p(x_{1:n}) l(x_{1:n}) = \frac{1}{n} El(x_{1:n}). \quad (302)$$

utilizando os comprimentos do código de Shannon, teremos

$$\log 1/p_i \leq l_i \leq \log 1/p_i + 1 \quad (303a)$$

$$\sum_i p_i \log 1/p_i \leq \sum_i p_i l_i \leq \sum_i p_i (\log 1/p_i + 1) \quad (303b)$$

$$H(X_1, \dots, X_n) \leq El(X_{1:n}) \leq H(X_1, \dots, X_n) + 1 \quad (303c)$$

$$H(X) \leq L_n \leq H(X) + \frac{1}{n}, \quad (303d)$$

onde utilizamos que  $X_i$  são i.i.d., então  $H(X_1, \dots, X_n) = nH(X_i)$ .

Quando  $n$  cresce, a penalidade por símbolo imposta ao código de Shannon diminui e conseguiremos assim aproximar-nos do limite da Entropia (por símbolo), embora teremos que adotar a estratégia de codificação em blocos.

A escolha feita na Equação (298) para definir o comprimento das palavras no código de Shannon, subentende o conhecimento da distribuição  $p$ . Entretanto, em geral, a distribuição subjacente não é conhecida. Isso implica que existem erros nos cálculos. Suponha então que o código de Shannon utiliza  $l(x) = \lceil \log 1/q(x) \rceil$ , mas a real distribuição é  $p(x) \neq q(x)$ . Vamos recalcular o valor esperado do comprimento do código, agora utilizando esta informação. Encontraremos novos limites para o comprimento esperado do código de Shannon.

**Teorema 39** (Limites para o código de Shannon com a distribuição aproximada) *O comprimento esperado de um código sob a distribuição  $p(x)$  com  $l(x) = \lceil \log 1/q(x) \rceil$  satisfaz*

$$H(p) + D(p \parallel q) \leq E_p l(X) \leq H(p) + D(p \parallel q) + 1. \quad (304)$$

*Demonstração.* Vamos analisar primeiramente o limite superior.

$$El(X) = \sum_x p(x) \lceil \log 1/q(x) \rceil \quad (305a)$$

$$\leq \sum_x p(x) \left( \log \frac{1}{q(x)} + 1 \right) \quad (305b)$$

$$= \sum_x p(x) \left( \log \frac{p(x)}{q(x)} \frac{1}{p(x)} + 1 \right) \quad (305c)$$

$$= \sum_x p(x) \log \frac{p(x)}{q(x)} + \sum_x p(x) \log \frac{1}{p(x)} + 1 \quad (305d)$$

$$= D(p \parallel q) + H(p) + 1. \quad (305e)$$

E agora, para o limite inferior temos

$$El(X) = \sum_x p(x) \lceil \log 1/q(x) \rceil \quad (306a)$$

$$\geq \sum_x p(x) \log 1/q(x) \quad (306b)$$

$$= \sum_x p(x) \left( \log \frac{p(x)}{q(x)} \frac{1}{p(x)} \right) \quad (306c)$$

$$= \sum_x p(x) \log \frac{p(x)}{q(x)} + \sum_x p(x) \log \frac{1}{p(x)} \quad (306d)$$

$$= D(p \parallel q) + H(p). \quad (306e)$$

□

Comparando estes limites com aqueles dados na Equação (301), verificamos que  $D(p \parallel q)$  é o custo adicional por símbolo causado por utilizar a distribuição errada.

Um aspecto importante na análise de códigos é entender como seus comprimentos de palavras se comparam a outros códigos univocamente decodificáveis em termos de eficiência. O teorema a seguir



aborda essa questão, fornecendo uma cota probabilística para a diferença entre os comprimentos de palavras do código de Shannon e de um código concorrente. Essa cota, expressa em função de uma constante  $c$ , destaca a robustez do código de Shannon, mostrando que a probabilidade de seus comprimentos excederem significativamente os de outro código decresce exponencialmente, um resultado que reforça sua relevância.

**Teorema 40** (Otimalidade Competitiva do Código de Shannon) *Considere  $l(x)$  como o comprimento da palavra associada ao símbolo  $x$  no código de Shannon e  $l'(x)$  como o comprimento da palavra correspondente em outro código univocamente decodificável, ambos definidos sobre uma variável aleatória  $X$  com distribuição  $p$ . Então, a probabilidade de que  $l(X)$  exceda  $l'(X)$  por pelo menos uma constante  $c$  é limitada por:*

$$\Pr(l(X) \geq l'(X) + c) \leq \frac{1}{2^{c-1}}. \quad (307)$$

*Equivalentemente, em notação mais formal:*

$$\Pr(\mathbf{1}_{\{l(X) \geq l'(X) + c\}} = 1) \leq \frac{1}{2^{c-1}}, \quad (308)$$

onde  $\mathbf{1}_{\{\cdot\}}$  é a função indicadora.

*Demonstração.*

$$\Pr(l(X) \geq l'(X) + c) = \Pr(\lceil \log 1/p(X) \rceil \geq l'(X) + c) \quad (309a)$$

$$\leq \Pr(\log 1/p(X) \geq l'(X) + c - 1) \quad (309b)$$

$$= \Pr(p(X) \leq 2^{-l'(X) - c + 1}) \quad (309c)$$

$$= \sum_{x: p(x) \leq 2^{-l'(x) - c + 1}} p(x) \quad (309d)$$

$$\leq \sum_{x: p(x) \leq 2^{-l'(x) - c + 1}} 2^{-l'(x) - c + 1} \quad (309e)$$

$$\leq \sum_x 2^{-l'(x)} 2^{-(c-1)} \leq 2^{-(c-1)}, \quad (309f)$$

onde em 309b utilizamos que o código de Shannon utiliza  $l(x) = \lceil \log 1/p(x) \rceil$ , e, em 309f, utilizamos a desigualdade de Kraft:  $\sum_x 2^{-l'(x)} \leq 1$ .

□

A probabilidade do código de Shannon ter comprimento esperado maior do que um outro código unicamente decodificável é exponencialmente decrescente com  $c > 1$ . Código de Shannon é ótimo para distribuições d-ádicas, já que neste caso teremos que  $\log 1/p(x)$  será inteiro.

Em distribuições  $d$ -ádicas, é mais provável que os comprimentos das palavras de um código de Shannon sejam menores do que maiores em comparação com um código alternativo. Esse resultado evidencia uma vantagem probabilística do código de Shannon, destacando sua eficiência relativa.

**Teorema 41** (Otimidade Competitiva do Código de Shannon II)

Considere uma distribuição  $d$ -ádica  $p$  sobre um conjunto de símbolos, onde  $p(x) = 1/d^k$  para algum inteiro  $k$ , e seja  $l(x) = \log_d(1/p(x))$  o comprimento da palavra associada a  $x$  no código de Shannon, enquanto  $l'(x)$  é o comprimento da palavra correspondente em outro código prefixo, ambos aplicados a uma variável aleatória  $X$  com distribuição  $p$ . Então:

$$\Pr(l(X) < l'(X)) \geq \Pr(l(X) > l'(X)), \quad (310)$$

com igualdade se, e somente se,  $l'(x) = l(x)$  para todo  $x$ .

*Demonstração.* Seja

$$\text{sign}(t) = \begin{cases} 1, & t > 0 \\ 0, & t = 0 \\ -1, & t < 0 \end{cases} \quad (311)$$

É fácil verificar que  $\text{sign}(t) \leq 2^t - 1$  para  $t = 0, \pm 1, \pm 2, \dots$ . Para  $t = 0$  teremos  $0 = 2^0 - 1 = 0$ ; para  $t \geq 1$  teremos  $1 \leq 2^t - 1$ , uma vez que  $2^t > 1$  para  $t \geq 1$ ; e para  $t \leq -1$  teremos  $-1 \leq 2^t - 1$ , uma vez que  $2^t < 1$  para  $t \leq -1$ .

Teremos então

$$\Pr(l'(X) < l(X)) - \Pr(l'(X) > l(X)) = \sum_{x: l'(x) \leq l(x)} p(x) - \sum_{x: l'(x) > l(x)} p(x) \quad (312a)$$

$$= \sum_x p(x) \text{sign}(l(x) - l'(x)) \quad (312b)$$

$$= E[\text{sign}(l(X) - l'(X))] \quad (312c)$$

$$\leq \sum_x p(x) (2^{l(x) - l'(x)} - 1) \quad (312d)$$

$$= \sum_x 2^{-l(x)} (2^{l(x) - l'(x)} - 1) \quad (312e)$$

$$= \sum_x 2^{-l'(x)} - \sum_x 2^{-l(x)} \quad (312f)$$

$$= \sum_x 2^{-l'(x)} - 1 \quad (312g)$$

$$\leq 1 - 1 \quad (312h)$$

$$= 0 \quad (312i)$$

onde em 312b utilizamos que  $\{l'(x) \leq l(x)\} \cap \{l'(x) > l(x)\} = \emptyset$ , em 312e utilizamos que  $p(x)$  é d-ádica,  $p(x) = 2^{-l(x)}$ , e em 312h utilizamos que  $l'(x)$  satisfaz Kraft. Assim, mostramos que  $\Pr(l'(X) < l(X)) \leq \Pr(l'(X) > l(X))$ , como desejado.  $\square$

### Desigualdade de Kraft revisitada

Considerando como objetivo principal, encontrar um código unívoco (unicamente decodificável) com comprimento esperado de palavra mínimo, ao observar as classes de códigos, conforme apresentadas na Figura 29, podemos imaginar que a classe mais ampla é a mais provável de se encontrar tal código desejado. Potencialmente, dentre os códigos unicamente decodificáveis (um conjunto maior), podemos encontrar um código com menor comprimento esperado. Entretanto, o teorema a seguir mostra que a desigualdade de Kraft é também condição *sine qua non* para obtermos um código unicamente decodificável. Desta forma, podemos concluir que um código prefixo é a melhor escolha.

Primeiramente, vamos apresentar o seguinte lemma.

**Lema 42** (Expansão Multinomial) *Seja  $\mathcal{X} = \{x_1, x_2, \dots, x_m\}$ ,  $|\mathcal{X}| = m$ , temos*

$$\left( \sum_{x \in \mathcal{X}} D^{-l(x)} \right)^k = \sum_{x_1, \dots, x_k \in \mathcal{X}^k} D^{-l(x_1)} D^{-l(x_2)} \dots D^{-l(x_k)}. \quad (313)$$

*Demonstração.* A demonstração utiliza o teorema multinomial para expandir a soma, revelando sua equivalência à soma combinatória das contribuições de cada sequência.

$$\left( \sum_{x \in \mathcal{X}} D^{-l(x)} \right)^k = \left( \sum_{i=1}^m D^{-l(x_i)} \right)^k \quad (314a)$$

$$\begin{aligned} &= \left( D^{-l(x_1)} + D^{-l(x_2)} + \dots + D^{-l(x_m)} \right)^k \\ &= \sum_{\substack{n_1, n_2, \dots, n_m \geq 0 \\ n_1 + n_2 + \dots + n_m = k}} \frac{k!}{n_1! n_2! \dots n_m!} D^{-l(x_1)n_1} \dots D^{-l(x_m)n_m} \end{aligned} \quad (314b)$$

$$= \sum_{x_1, \dots, x_k \in \mathcal{X}^k} D^{-l(x_1)} D^{-l(x_2)} \dots D^{-l(x_k)}. \quad (314c)$$

Cada termo  $(D^{-l(x_1)})^{n_1} (D^{-l(x_2)})^{n_2} \dots (D^{-l(x_m)})^{n_m}$  pode ser visto como o produto de  $k$  fatores  $D^{-l(x_i)}$ , onde  $x_i$  é escolhido  $n_i$  vezes. Isso corresponde exatamente a somar  $D^{-l(x_1)} D^{-l(x_2)} \dots D^{-l(x_k)}$  sobre todas as sequências possíveis  $x_{1:k} = (x_1, x_2, \dots, x_k)$  em  $\mathcal{X}^k$ :

$$\sum_{x_{1:k} \in \mathcal{X}^k} D^{-l(x_1)} D^{-l(x_2)} \dots D^{-l(x_k)}$$

Para cada sequência  $x_{1:k}$ , o produto reflete uma combinação específica dos termos  $D^{-l(x_i)}$ , e a soma cobre todas as  $m^k$  possibilidades.  $\square$

**Teorema 43** (Desigualdade de Kraft para códigos unicamente decodificáveis) *O comprimento de palavras de qualquer código unicamente decodificável (não necessariamente instantâneo) deve satisfazer a desigualdade de Kraft  $\sum_i D^{-l_i} \leq 1$ . Por outro lado, dado um conjunto de comprimentos que satisfazem Kraft, é possível construir um código unicamente decodificável.*

*Demonstração.* A proposição inversa já foi previamente mostrada, uma vez que mostramos como construir um código instantâneo utilizando um conjunto de comprimentos satisfazendo Kraft.

Dado um código unicamente decodificável (não necessariamente instantâneo) com comprimentos  $l(x)$  e com extensão com comprimentos dados por  $l(x_1, \dots, x_k) = \sum_{i=1}^k l(x_i)$ , queremos mostrar que  $\sum_x D^{-l(x)} \leq 1$ . Vamos definir  $S = \sum_{x \in \mathcal{X}} D^{-l(x)}$  e então teremos

$$S^k = \left( \sum_{x \in \mathcal{X}} D^{-l(x)} \right)^k \quad (315a)$$

$$= \sum_{x_{1:k} \in \mathcal{X}^k} D^{-l(x_1)} D^{-l(x_2)} \dots D^{-l(x_k)} \quad (315b)$$

$$= \sum_{x_{1:k} \in \mathcal{X}^k} D^{-(\sum_{i=1}^k l(x_i))} \quad (315c)$$

$$= \sum_{x_{1:k} \in \mathcal{X}^k} D^{-l(x_{1:k})} \quad (315d)$$

$$= \sum_{m=1}^{kl_{\max}} a(m) D^{-m} \quad (315e)$$

onde em 315b utilizamos o resultado do Teorema 42 e  $l_{\max} = \max_x l(x)$ ,  $a(m)$  é o número de sequências  $x_{1:k}$  mapeadas em palavras de comprimento  $m$ , i.e.,

$$a(m) = |\{x_{1:k} \in \mathcal{X}^k : l(x_{1:k}) = m\}|. \quad (316)$$

Existem  $D^m$  palavras de comprimento  $m$ , e cada uma delas pode ter no máximo uma sequência da fonte associada, já que o código é

unicamente decodificável. Então  $a(m) \leq D^m$ , e assim

$$\underbrace{S^k}_{\text{exponencial em } k} = \sum_{m=1}^{kl_{\max}} a(m) D^{-m} \quad (317a)$$

$$\leq \sum_{m=1}^{kl_{\max}} D^m D^{-m} = \underbrace{kl_{\max}}_{\text{polinomial em } k} \quad \forall k \quad (317b)$$

Isto só pode ser verdade para todo  $k$  se  $S \leq 1$ . Teremos então

$$S = \sum_{x \in \mathcal{X}} D^{-l(x)} \leq 1. \quad (318)$$

□

Concluimos assim que todo código unicamente decodificável deve satisfazer a desigualdade de Kraft. Sabemos também que é possível construir um código instantâneo com os comprimentos que satisfazem Kraft. Logo, não há razão em se buscar um código unicamente decodificável que não seja um código de prefixo, afinal para um mesmo comprimento esperado podemos construir um código de prefixo.

### *Código de Huffman*

Os códigos de comprimento variável buscam minimizar o comprimento esperado, para tanto, atribuem comprimentos menores a símbolos mais frequentes em detrimento de códigos maiores para símbolos menos frequentes.

Antes de 1952, as abordagens predominantes para esses códigos eram *top-down*, partindo de uma estrutura inicial e ajustando comprimentos de forma descendente, mas nem sempre resultavam em soluções ótimas. David A. Huffman, em 1952, utilizou uma abordagem inovadora: construir a árvore binária de maneira *bottom-up*, combinando iterativamente os símbolos de menor probabilidade. Esse método garante um código prefixo com comprimento esperado mínimo, respeitando a desigualdade de Kraft e aproximando-se do limite teórico da entropia de Shannon.

Vejamos inicialmente um exemplo de utilização da abordagem gulosa *top-down* para criar um código. A abordagem consiste em: começar pelo topo e realizar divisões com entropia máxima para determinar os símbolos. Em cada etapa dividimos os símbolos em dois grupos, de forma que cada etapa tenha entropia máxima no processo de determinação dos símbolos.

**Exemplo 22** (Método Guloso *top-down* para codificação (Bilmes 2013)). Considere uma variável aleatória  $X$  com distribuição

	a	b	c	d	e	f	g
p	0.01	0.24	0.05	0.20	0.47	0.01	0.02

Em cada etapa, queremos dividir os símbolos em dois grupos de forma a maximizar a entropia nesta divisão. A pergunta  $Y_1$  para determinação (ou codificação) de  $X$ , é aquela que reduz a entropia residual sobre  $X$ .

$$H(X|Y_1) = H(X, Y_1) - H(Y_1) \quad (319a)$$

$$= H(X) - H(Y_1) \quad (319b)$$

onde utilizamos o fato de que  $Y_1$  é uma função de  $X$ , e assim  $H(X, Y_1) = H(X)$ . Escolhemos a pergunta  $Y_1$  com maior informação mútua com  $X$ :

$$I(Y_1; X) = H(X) - H(X|Y_1) = H(Y_1), \quad (320)$$

e desta forma buscaremos minimizar a incerteza residual sobre  $X$ , dado  $Y_1$ .

Considere a seguinte partição  $\{a, b, c, d, e, f, g\} = \{a, b, c, d\} \cup \{e, f, g\}$ . Com este particionamento, obteremos entropia máxima pois  $p(X \in \{a, b, c, d\}) = p(X \in \{e, f, g\}) = 0.5$ . Este particionamento corresponde à v.a.  $Y_1 = \mathbf{1}_{\{X \in \{e, f, g\}\}}$ , e teremos  $H(Y_1) = 1$ , valor máximo, reduzindo assim  $H(X|Y_1)$ .

Para o caso  $Y_1 = 0$ , podemos agora particionar  $\{a, b, c, d\} = \{a, b\} \cup \{c, d\}$ , uma vez que  $p(\{a, b\}) = p(\{c, d\}) = 1/4$ . Desta forma,  $p(X \in \{a, b\} | X \in \{a, b, c, d\}) = 1/2$  e  $p(X \in \{c, d\} | X \in \{a, b, c, d\}) = 1/2$ . Este particionamento corresponde a  $Y_2 = \mathbf{1}_{\{X \in \{c, d\}\}}$ . Obtemos assim  $H(Y_2 | Y_1 = 0) = 1$ , novamente alcançando o máximo.

Para o caso  $Y_1 = 1$  precisaremos particionar  $\{e, f, g\}$ . As opções são listadas a seguir:

	I	II	III
partição	$(\{e\}, \{f, g\})$	$(\{e, f\}, \{g\})$	$(\{e, g\}, \{f\})$
prob	$(0.47, 0.03)$	$(0.48, 0.02)$	$(0.49, 0.01)$
$H(Y_2   Y_1 = 1)$	0.3274	0.2423	0.1414

A partição I é aquela que fornece maior entropia.

Tais escolhas nos levam a maximizar  $H(Y_2 | Y_1)$  e por conseguinte iremos minimizar a incerteza remanescente sobre  $X$ :

$$H(X | Y_2, Y_1) = H(X, Y_2 | Y_1) - H(Y_2 | Y_1) \quad (321a)$$

$$= H(X | Y_1) - H(Y_2 | Y_1) \quad (321b)$$

$$= H(X) - H(Y_2 | Y_1) - H(Y_1). \quad (321c)$$

A cada passo escolhemos o que nos parece melhor, as escolhas futuras terão que lidar com as possibilidades que restaram devido às escolhas passadas. Teremos assim os seguintes particionamentos:

conjunto	partição	probabilidades	entropia condicional
$\{a, b, c, d, e, f, g\}$	$\{a, b, c, d\}, \{e, f, g\}$	$(0.5, 0.5)$	$H(Y_1) = 1$
$\{a, b, c, d\}$	$\{a, b\}, \{c, d\}$	$(0.25, 0.25)$	$H(Y_2   Y_1 = 0) = 1$
$\{e, f, g\}$	$\{e\}, \{f, g\}$	$(0.47, 0.03)$	$H(Y_2   Y_1 = 1) = 0.3274$
$\{a, b\}$	$\{a\}, \{b\}$	$(0.01, 0.24)$	$H(Y_3   Y_2 = 0, Y_1 = 0) = 0.2423$
$\{c, d\}$	$\{c\}, \{d\}$	$(0.05, 0.20)$	$H(Y_3   Y_2 = 1, Y_1 = 0) = 0.7219$
$\{e\}$	$\{e\}$	$(0.47)$	$H(Y_3   Y_2 = 0, Y_1 = 1) = 0$
$\{f, g\}$	$\{f\}, \{g\}$	$(0.01, 0.02)$	$H(Y_3   Y_2 = 1, Y_1 = 1) = 0.9183$

Ao final, seguindo a abordagem gulosa *top-down* obtemos o código representado na Figura 31. Este código possui comprimento esperado do código  $El = 2.53$  e eficiência  $H/El = 0.7638$ , uma vez que a entropia é  $H = 1.9323$  bits.

A título de comparação, o algoritmo de Huffman produz o código apresentado na Figura 32. Terá comprimento esperado  $El_{\text{huffman}} = 1.97$  e eficiência  $H/El_{\text{huffman}} = 0.9809$ .

Isto demonstra claramente que o código de Huffman é superior e que a estratégia gulosa não foi capaz de nos levar a um código ótimo.

O algoritmo de Huffman pode ser sumarizado através dos seguintes passos:

1. Selecione os dois símbolos menos prováveis no alfabeto.
2. Ambos terão associados as palavras mais longas e se diferirão no último bit.
3. Combine estes dois símbolos em um símbolo auxiliar com probabilidade igual à soma das probabilidades dos dois símbolos. Adicione o símbolo auxiliar e remova os dois símbolos previamente selecionados. Repita os passos.

O procedimento é similar para  $D > 2$ . Neste caso poderá ser necessário utilizar símbolos fictícios no alfabeto.

**Exemplo 23** ((Bilmes 2013)). Seja  $\mathcal{X} = \{1, 2, 3, 4, 5\}$  com probabilidades  $(1/4, 1/4, 1/5, 3/20, 3/20)$ . O procedimento e o código de Huffman obtido estão dispostos na Figura 33.

$\log \frac{1}{p(x)}$	length	codeword	$\chi$	prob	step 1	prob	step 2	prob	step 3	prob	step 4	prob
2.0	2	00	1	0.25	—	0.25	—	0.25	0	0.55	0	1.0
2.0	2	10	2	0.25	—	0.25	0	0.45	—	0.45	1	
2.3	2	11	3	0.2	—	0.2	1	—	—	—	—	
2.7	3	010	4	0.15	0	0.3	—	0.3	1	—	—	
2.7	3	011	5	0.15	1	—	—	—	—	—	—	

Para este código de Huffman teremos  $El = 2.3$  bits e  $H = 2.2855$  bits. Note que, algumas palavras são maiores e outras menores que  $l^*(x) = I(X) = \log 1/p(x)$ . O código de Huffman encontrado pode ser expresso, de maneira compacta, como:  $((2, 3), (1, (4, 5)))$ , onde cada par de parênteses representa os símbolos combinados em cada passo do algoritmo.

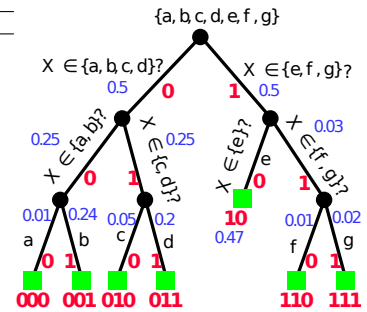


Figura 31: Árvore obtida através do método guloso seguindo a estratégia *top-down*.

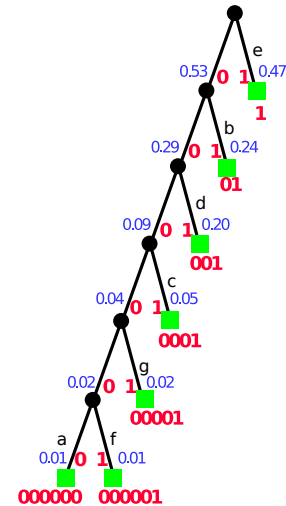


Figura 32: Árvore obtida pelo código de Huffman.

Figura 33: Código de Huffman exemplo.

O lema a seguir descreve atributos necessários para códigos de prefixo ótimos, relacionando a frequência dos símbolos à estrutura de suas palavras. Ele assegura que símbolos mais prováveis tenham comprimentos menores ou iguais e que as duas palavras mais longas, ligadas aos símbolos menos frequentes, possuam comprimentos iguais e difiram apenas no último bit. Essas propriedades, intrínsecas a códigos ótimos, refletem a construção *bottom-up* do algoritmo de Huffman.

**Lema 44** (Propriedades de Códigos Instantâneos Ótimos) *Para toda distribuição,  $\exists$  um código instantâneo ótimo (i.e., com comprimento esperado mínimo) satisfazendo simultaneamente:*

1. *Se  $p_j > p_k$  então  $l_j \leq l_k$  (i.e., o símbolo mais provável não possui palavra com maior comprimento).*
2. *As duas maiores palavras possuem o mesmo comprimento.*
3. *As duas maiores palavras diferem apenas no último bit e correspondem aos dois símbolos menos prováveis.*

*Demonstração.* Vejamos primeiramente a propriedade 1.

Suponha que  $C_m$  seja um código ótimo (então  $L(C_m)$  é mínimo) e escolha  $j, k$  de forma tal que  $p_j > p_k$ . Precisamos mostrar que  $\exists$  um código com  $l_j \leq l_k$ . Considere o código  $C'_m$  com a troca das palavras  $j$  e  $k$ , ou seja,

$$l'_j = l_k \quad \text{e} \quad l'_k = l_j, \quad (322)$$

o que só pode tornar o código maior, então  $L(C'_m) \geq L(C_m)$ .

Realizando a troca, como  $L(C_m)$  é mínimo, teremos

$$0 \leq L(C'_m) - L(C_m) = \sum_i p_i l'_i - \sum_i p_i l_i \quad (323a)$$

$$= p_j l'_j + p_k l'_k - p_j l_j - p_k l_k = p_j l_k + p_k l_j - p_j l_j - p_k l_k \quad (323b)$$

$$= p_j (l_k - l_j) - p_k (l_k - l_j) \quad (323c)$$

$$= \underbrace{(p_j - p_k)}_{>0} (l_k - l_j) \quad (323d)$$

Devemos ter então  $(l_k - l_j) \geq 0$ , ou seja,  $l_k \geq l_j$  quando  $p_j > p_k$ , satisfazendo assim a propriedade 1. Esta propriedade é verdadeira para todos códigos ótimos.

Vejamos agora a propriedade 2: as palavras mais longas possuem o mesmo comprimento.

Se as duas palavras maiores não possuem o mesmo comprimento, podemos apagar o último bit da palavra mais longa. Desta forma manteremos a propriedade de prefixo, já que a palavra mais longa é a única com o seu dado comprimento e não existe um prefixo dela que seja uma outra palavra (veja a Figura 34).



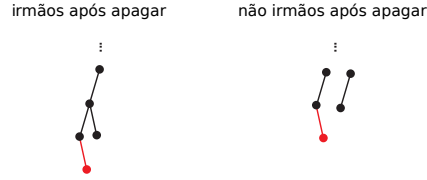


Figura 34: Apagando o último bit da palavra mais longa.

Desta forma reduzimos o comprimento esperado. Concluimos que um código ótimo deve possuir as duas palavras mais longas com o mesmo comprimento.

Analisaremos agora a propriedade 3: as duas palavras mais longas diferem apenas no último bit e correspondem aos símbolos menos prováveis.

Devido à propriedade 1 ( $p_k < p_j \Rightarrow l_k \geq l_j$ ), se  $p_k$  é a menor probabilidade, então ela deve possuir associada uma palavra de comprimento que não seja menor do que qualquer outra  $j$  com  $p_j > p_k$ . De forma similar, se  $p_k$  é a segunda menor probabilidade, a palavra associada deve possuir comprimento que não seja menor do que qualquer outro símbolo mais provável. As duas palavras mais longas possuem o mesmo comprimento (propriedade 2) e correspondem aos dois símbolos menos prováveis. Se as duas maiores palavras não são irmãs, podemos trocá-las. Isto é, se  $p_1 \geq p_2 \geq \dots \geq p_m$ , então fazemos a transposição ilustrada na Figura 35.

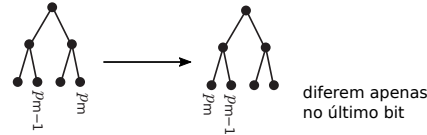


Figura 35: Transposição das duas palavras mais longas.

Isto não altera o comprimento esperado  $L = \sum_i p_i l_i$ .

□

O Teorema 44 nos fornece que, se  $p_1 \geq p_2 \geq \dots \geq p_m$ , então existe um código ótimo com  $l_1 \leq l_2 \leq \dots \leq l_{m-1} = l_m$  e no qual  $C(x_{m-1})$  e  $C(x_m)$  diferem apenas no último bit.

Vamos mostrar que Huffman é ótimo através da operação de criar um novo código no qual a otimização é mais simples. Este processo é repetido até que a otimização seja trivial.

**Teorema 45** (Otimidade do Código de Huffman) *O procedimento de codificação de Huffman cria um código com comprimentos inteiros ótimo.*

*Demonstração.* Assuma um código  $C_m$  (não necessariamente ótimo) que satisfaz as propriedades anteriores.  $C_m$  terá as seguintes palavras de código  $\{w_i\}_{i=1}^m$ . O procedimento de Huffman consiste em transformar  $C_m$  em  $C_{m-1}$ , com palavras de código  $\{w'_i\}_{i=1}^{m-1}$ . Para o código

$C_m$  teremos as seguintes palavras, comprimento e probabilidades associados:

$C_m$	comprimento	prob. do símbolo
$w_1$	$l_1$	$p_1$
$w_2$	$l_2$	$p_2$
$\vdots$	$\vdots$	$\vdots$
$w_{m-2}$	$l_{m-2}$	$p_{m-2}$
$w_{m-1}$	$l_{m-1}$	$p_{m-1}$
$w_m$	$l_m$	$p_m$

Para ir de  $C_m$  para  $C_{m-1}$ , o procedimento de Huffman combina dos dois símbolos de menor probabilidades em  $C_m$ , criando um símbolo auxiliar em  $C_{m-1}$  com probabilidade dada pela soma  $p'_{m-1} = p_{m-1} + p_m$ . A transformação de  $C_m$  em  $C_{m-1}$  é esquematizada a seguir:

prob. símb.	$C_{m-1}$	comp. $m-1$	rel. código	rel. comp.	prob. símb.
$p_1$	$w'_1$	$l'_1$	$w_1 = w'_1$	$l_1 = l'_1$	$p_1$
$p_2$	$w'_2$	$l'_2$	$w_2 = w'_2$	$l_2 = l'_2$	$p_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$p_{m-2}$	$w'_{m-2}$	$l'_{m-2}$	$w_{m-2} = w'_{m-2}$	$l_{m-2} = l'_{m-2}$	$p_{m-2}$
$p_{m-1} + p_m$	$w'_{m-1}$	$l'_{m-1}$	$w_{m-1} = w'_{m-1}0$ $w_m = w'_{m-1}1$	$l_{m-1} = l'_{m-1} + 1$ $l_m = l'_{m-1} + 1$	$p_{m-1}$ $p_m$

onde  $w_i$  e  $l_i$  são as palavras e comprimento das palavras do código  $C_m$  e  $w'_i$  e  $l'_i$  do código  $C_{m-1}$ .

As palavras e comprimentos em Huffman são definidos recursivamente. Huffman define uma relação entre palavras (e consequentemente comprimentos) ao dar um passo de um código para outro mais simples.

Obtemos assim a seguinte relação entre o comprimento esperado de  $C_m$  e o comprimento esperado de  $C_{m-1}$ :

$$L(C_m) = \sum_i p_i l_i \quad (324a)$$

$$= \sum_{i=1}^{m-2} p_i l'_i + p_{m-1} (l'_{m-1} + 1) + p_m (l'_{m-1} + 1) \quad (324b)$$

$$= \sum_{i=1}^{m-2} p_i l'_i + (p_{m-1} + p_m) l'_{m-1} + p_{m-1} + p_m \quad (324c)$$

$$= \sum_{i=1}^{m-1} p'_i l'_i + p_{m-1} + p_m \quad (324d)$$

$$= L(C_{m-1}) + p_{m-1} + p_m \quad (324e)$$

onde o termo  $p_{m-1} + p_m$  é constante, não envolve comprimentos de palavras e, conseqüentemente, não tem nenhuma implicância no processo de otimização.

Desta forma, reduzimos o número de variáveis que iremos otimizar de  $m$  para  $m - 1$ . O procedimento de Huffman implica em

$$\min_{l_{1:m}} L(C_m) = \text{const.} + \min_{l_{1:m-1}} L(C_{m-1}) \quad (325a)$$

$$= \text{const.} + \min_{l_{1:m-2}} L(C_{m-2}) \quad (325b)$$

$$\vdots$$

$$= \text{const.} + \min_{l_{1:2}} L(C_2) \quad (325c)$$

onde em cada passo são preservadas as propriedades.

Ao reduzir para o caso com dois comprimentos ( $C_2$ ), teremos a solução óbvia, um bit para cada símbolo, e assim podemos refazer o caminho de volta e construir o código. Em cada passo garantimos que estaremos obtendo a solução ótima, e assim o código criado pelo algoritmo de Huffman será ótimo.  $\square$

### *Codificação Shannon-Fano-Elias*

A codificação Shannon-Fano-Elias é uma abordagem prática para representar símbolos com base em suas probabilidades cumulativas. Ela atribui códigos prefixo aproximando os comprimentos ideais por meio de uma representação binária truncada da função de distribuição cumulativa. Historicamente, Shannon-Fano-Elias é um marco na evolução da codificação, sendo ela predecessora da codificação aritmética, que também usa probabilidades cumulativas.

Vejamos algumas definições que serão utilizadas na codificação Shannon-Fano-Elias.

**Definição 42** (Distribuição Cumulativa). Seja  $\mathcal{X} = \{x_1, x_2, \dots, x_m\}$  um conjunto finito de símbolos com uma distribuição  $p$ , onde  $p(x_i) \geq 0$  e  $\sum_{i=1}^m p(x_i) = 1$ . A função de distribuição cumulativa, denotada por  $F(x)$ , é definida como:

$$F(x) = \sum_{a \leq x} p(a), \quad (326)$$

em que a soma é tomada sobre todos os símbolos  $a \in \mathcal{X}$  tais que  $a \leq x$ , considerando a ordenação de  $\mathcal{X}$ :  $x_1 < x_2 < \dots < x_m$ . Assim,  $F(x)$  representa a probabilidade acumulada de todos os símbolos até  $x$ , inclusive, e satisfaz as propriedades:  $0 \leq F(x) \leq 1$ ,  $F(x_i) \leq F(x_j)$  se  $x_i < x_j$  (monotonicidade), e  $F(x_m) = 1$ .

**Definição 43** ( $\bar{F}$ -Distribuição Cumulativa Modificada). Seja  $\mathcal{X} = \{x_1, x_2, \dots, x_m\}$  um conjunto finito de símbolos com uma distribui-

ção  $p$ , e seja  $F(x) = \sum_{a \leq x} p(a)$  a função de distribuição cumulativa associada, considerando uma ordenação fixa  $x_1 < x_2 < \dots < x_m$ . A função  $\bar{F}(x)$ , chamada de distribuição cumulativa modificada, é definida como:

$$\bar{F}(x) \triangleq \sum_{a < x} p(a) + \frac{1}{2}p(x), \quad (327a)$$

$$= F(x) - \frac{1}{2}p(x), \quad (327b)$$

em que  $\sum_{a < x} p(a)$  é a soma das probabilidades dos símbolos estritamente menores que  $x$ , e  $\frac{1}{2}p(x)$  adiciona metade da probabilidade do símbolo  $x$ . Assim,  $\bar{F}(x)$  representa um ponto intermediário no intervalo de probabilidade associado a  $x$  na distribuição acumulada.

$\bar{F}(x)$  é o ponto entre  $F(x-1)$  e  $F(x)$ , então, como  $p(x) > 0$ , temos

$$F(x-1) < \bar{F}(x) < F(x) \quad (328)$$

Como  $p(x) > 0$ ,  $a \neq b \Rightarrow F(a) \neq F(b) \Leftrightarrow \bar{F}(a) \neq \bar{F}(b)$ . Podemos utilizar  $\bar{F}(a)$  como um código não singular para  $a$  (podemos utilizar a expansão binária após a vírgula, como visto anteriormente na prova de Kraft para um infinito contável de comprimentos). Teremos um código unicamente decodificável, porém poderemos ter palavras de tamanho infinito. A solução será truncar  $\bar{F}(x)$  para ficar com  $l(x)$  bits. A notação para tanto será  $\lfloor \bar{F}(x) \rfloor_{l(x)}$ .

**Exemplo 24** (Truncamento com  $l$  bits). Seja  $l = 4$  e  $\bar{F}(x) = 0.011011001000\dots$ , então

$$\lfloor \bar{F}(x) \rfloor_{l(x)} = 0.0110. \quad (329)$$

	0.0110	1100	1000	
–	0.0110	0000	0000	código $\lfloor \bar{F}(x) \rfloor_4$
=	0.0000	1100	1000	
<	0.0001	0000	0000	$= 1/2^{l(x)}$

Precisamos agora determinar qual seria  $l(x)$  mínimo para que a decodificação unívoca seja preservada. Sabemos que

$$\bar{F}(x) - \lfloor \bar{F}(x) \rfloor_{l(x)} < \frac{1}{2^{l(x)}}, \quad (330)$$

conforme ilustrado no Exemplo 24. Se considerarmos  $l(x) = \lceil \log 1/p(x) \rceil + 1$ , então teremos

$$\frac{1}{2^{l(x)}} = \frac{1}{2} 2^{-\lceil \log 1/p(x) \rceil} \leq \frac{1}{2} 2^{-\log 1/p(x)} = \frac{p(x)}{2} \quad (331a)$$

$$= \bar{F}(x) - F(x-1) \quad (331b)$$

Combinando com o limite anterior, teremos

$$\bar{F}(x) - \lfloor \bar{F}(x) \rfloor_{l(x)} < \frac{1}{2^{l(x)}} < \bar{F}(x) - F(x-1), \quad (332)$$

e poderemos concluir que

$$\lfloor \bar{F}(x) \rfloor_{l(x)} > F(x-1). \quad (333)$$

Por fim, temos

$$F(x-1) < \lfloor \bar{F}(x) \rfloor_{l(x)} \leq \bar{F}(x) < F(x). \quad (334)$$

Concluimos que  $l(x) = \lceil \log 1/p(x) \rceil + 1$  bits serão suficientes para descrever  $x$  de forma não ambígua segundo a representação  $\lfloor \bar{F}(x) \rfloor_{l(x)}$ , uma vez que para cada  $x$  teremos  $\lfloor \bar{F}(x) \rfloor_{l(x)}$  em intervalos distintos, conforme a desigualdade anterior.

Precisamos agora verificar que a representação binária do  $\lfloor \bar{F}(x) \rfloor_{l(x)}$  nos fornecerá um código de prefixo. Para tanto, considere a palavra  $z_1 z_2 \dots z_l$  que corresponde ao intervalo semiaberto:

$$\underbrace{\lfloor \bar{F}(x) \rfloor_{l(x)}}_{[0.z_1 z_2 \dots z_l, \underbrace{\lfloor \bar{F}(x) \rfloor_{l(x)}}_{0.z_1 z_2 \dots z_l + 1/2^l})} = [0.z_1 z_2 \dots z_l, 0.z_1 z_2 \dots z_l + 0.00 \dots 1). \quad (335)$$

que possui comprimento  $1/2^l$  (este intervalo contém todos os números binários que se iniciam com  $0.z_1 z_2 \dots z_l$ ).

A desigualdade  $F(x-1) < \lfloor \bar{F}(x) \rfloor_{l(x)} \leq \bar{F}(x) < F(x)$  e o intervalo de comprimento  $1/2^l$  são representados na Figura 36

Então  $\lfloor \bar{F}(x) \rfloor_{l(x)} \in (F(x-1), \bar{F}(x)]$ . Como  $2^{-l(x)} \leq p(x)/2 = \bar{F}(x) - F(x-1)$  e também  $F(x-1) < \lfloor \bar{F}(x) \rfloor_{l(x)} \leq \bar{F}(x) < F(x)$ , então teremos que os intervalos abertos são disjuntos, mesmo se  $\lfloor \bar{F}(x) \rfloor_{l(x)} = \bar{F}(x)$ . A forma de codificação proposta, usando a representação binária de  $\lfloor \bar{F}(x) \rfloor_{l(x)}$ , nos leva a um código de prefixo (não existirão duas palavras associadas a um mesmo intervalo e os intervalos associados a cada um são disjuntos). Como é suficiente ter  $l(x) = \lceil \log 1/p(x) \rceil + 1$ , podemos calcular o limite para o comprimento esperado. Assim,

$$L = \sum_x p(x) l(x) = \sum_x p(x) (\lceil \log 1/p(x) \rceil + 1) \leq H(X) + 2 \quad (336)$$

**Exemplo 25** (Distribuição d-ádica). Considere a seguinte distribuição d-ádica. Os passos para encontrar o código de Shannon-Fano-Elias são explicitados abaixo.

$x$	$p(x)$	$F(x)$	$\bar{F}(x)$	$\bar{F}(x)$ (binário)	$l(x)$	palavra código
1	0.25	0.25	0.125	0.001	3	001
2	0.5	0.75	0.5	0.10	2	10
3	0.125	0.875	0.8125	0.1101	4	1101
4	0.125	1.0	0.9375	0.1111	4	1111

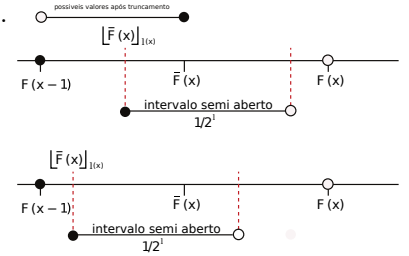


Figura 36: Intervalo de comprimento  $1/2^l$  representando  $\lfloor \bar{F}(x) \rfloor_{l(x)}$  como uma forma não ambígua para codificação.

Para o código encontrado teremos  $El = 2.75$  bits enquanto  $H = 1.75$  bits. Comparativamente, para o caso de Huffman teremos a árvore  $((3, 4), 1), 2)$ , e assim  $El_{\text{huffman}} = 0.25 + 0.25 \times 2 + 0.125 \times 3 + 0.125 \times 3 = 1.75$ .

**Exemplo 26** (Dízima periódica). Utilizaremos a seguinte notação para dízima periódica:  $0.\overline{01} = 0.0101010101\dots$ . Para uma dada distribuição  $p$ , abaixo apresentamos os passos para encontrar o código de Shannon-Fano-Elias:

$x$	$p(x)$	$F(x)$	$\overline{F}(x)$	$\overline{F}(x)$ (binário)	$l(x)$	palavra código
1	0.25	0.25	0.125	0.001	3	001
2	0.25	0.5	0.375	0.011	3	011
3	0.2	0.7	0.6	0.10011	4	1001
4	0.15	0.85	0.775	0.1100011	4	1100
5	0.15	1	0.925	0.1110110	4	1110

Para o código de Shannon-Fano-Elias encontrado, teremos  $H = 2.285$  bits,  $El = 3.5$  bits, enquanto  $El_{\text{huffman}} = 2.3$  bits, sendo a árvore de Huffman  $((1, (4, 5)), (3, 2))$ .

### *O Jogo de Shannon*

A compreensão da entropia como medida da incerteza em uma fonte de informação é fundamental para o desenvolvimento de códigos eficientes. Em seu artigo de 1951 (Shannon 1951), “*Prediction and Entropy of Printed English*”, Claude Shannon explorou essa ideia por meio de um experimento simples, conhecido como o Jogo de Shannon. Neste jogo, participantes tentam adivinhar a próxima letra em uma sequência de texto em inglês, como completar “TH” em “THE” ou “THEN”, utilizando o contexto das letras anteriores. Shannon observou que, em média, os participantes precisavam de poucas tentativas para acertar, sugerindo que a previsibilidade do idioma reduz significativamente a quantidade de informação por letra.

Para quantificar isso, Shannon modelou o texto como um processo estocástico e estimou a entropia condicional — a incerteza média de uma letra dado o contexto anterior. Usando amostras de até 100 letras e aproximações estatísticas (de monogramas, bigramas e trigramas), ele concluiu que a entropia do inglês impresso está entre 0,6 e 1,3 bits por letra, muito abaixo dos 4,76 bits de um alfabeto de 27 símbolos (26 letras mais espaço) com distribuição uniforme. Esse resultado implica uma redundância de cerca de 75%, evidenciando que a estrutura linguística permite compressão eficiente.

O Jogo de Shannon destaca a dependência entre símbolos em uma sequência. Por exemplo, após “QU”, a próxima letra é quase certamente uma vogal, reduzindo a incerteza. Essa visão sequencial motiva

alguns métodos, como a codificação aritmética, que comprime uma mensagem inteira em um único número, aproveitando a entropia de toda a sequência.

### *Codificação Aritmética*

A codificação aritmética supera as limitações dos códigos baseados em símbolos individuais, sendo considerada como um código de fluxo, ou seja, codifica uma sequência inteira em um único número real/binário. Ao mapear a mensagem em um intervalo de probabilidade que se estreita iterativamente com base nas probabilidades cumulativas dos símbolos, a codificação aritmética alcança taxas de compressão que se aproximam da entropia da fonte, mesmo em distribuições complexas com dependências entre símbolos. O algoritmo básico da codificação aritmética foi desenvolvido independentemente por Jorma Rissanen e Richard C. Pasco, na década de 1970. A ideia reflete uma evolução natural das propostas de Shannon, oferecendo uma solução elegante para fontes onde a redundância sequencial desempenha um papel importante.

A codificação aritmética opera dividindo subsequentemente um intervalo inicial de probabilidade,  $[0, 1)$ , com base nas probabilidades dos símbolos de uma sequência. Para cada novo símbolo, o intervalo atual é particionado proporcionalmente às probabilidades cumulativas dos símbolos possíveis, e o subintervalo correspondente ao símbolo observado é selecionado. À medida que a sequência avança, esse processo gera intervalos cada vez menores, cuja largura reflete a probabilidade conjunta da sequência inteira. Para transmitir a mensagem, escolhe-se um número binário que caiba dentro do intervalo final — tipicamente, uma sequência binária cuja representação decimal esteja contida nesse intervalo reduzido. Esse número único codifica toda a sequência.

Suponha que o alfabeto da fonte seja  $\mathcal{X} = \{a_1, a_2, \dots, a_I\}$ . À medida que a fonte produz uma sequência de símbolos  $x_1, x_2, \dots, x_n, \dots$ , o intervalo real  $[0, 1)$  é subsequentemente subdividido de acordo com as probabilidades dos símbolos em cada instante. A Figura 37 apresenta a subdivisão do intervalo para os dois primeiros símbolos da sequência.

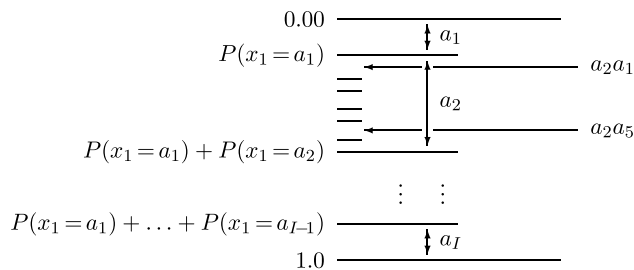


Figura 37: Definição dos intervalos probabilísticos para uma sequência de símbolos produzida pela fonte.

Paralelamente, o codificador deve ir subdividindo o intervalo  $[0, 1)$ , sempre ao meio em cada passo, para determinar uma sequência binária correspondente a um intervalo que esteja dentro do intervalo selecionado pela sequência de símbolos produzida pela fonte. A Figura 38 apresenta a subdivisão binária dos intervalos.

**Exemplo 27** ((MacKay 2003)). Suponha  $x \in \{a, b, \square\}$ , onde  $\square$  é o símbolo de término. Vamos considerar a codificação da *string*  $bbba\square$ . A seguir, apresentamos as probabilidades do próximo símbolo, dado o passado:

-	$p(a) = 0.425$	$p(b) = 0.425$	$p(\square) = 0.15$
$b$	$p(a b) = 0.28$	$p(b b) = 0.57$	$p(\square b) = 0.15$
$bb$	$p(a bb) = 0.21$	$p(b bb) = 0.64$	$p(\square bb) = 0.15$
$bbb$	$p(a bbb) = 0.17$	$p(b bbb) = 0.68$	$p(\square bbb) = 0.15$
$bbba$	$p(a bbba) = 0.28$	$p(b bbba) = 0.57$	$p(\square bbba) = 0.15$

Temos subintervalos semiabertos em  $[0, 1)$ . Os intervalos  $[l, u)$  são dados pela probabilidade condicional  $p(x_i|x_1, x_2, \dots, x_{i-1})$ . Temos a representação de um prefixo  $b_1b_2 \dots b_k$  por um intervalo da forma  $[0.b_1b_2 \dots b_k, 0.b_1b_2 \dots b_k + 2^{-k})$  para  $b_i \in \{0, 1\}$ .

A Figura 39 ilustra os intervalos que teremos ao final da sequência  $bbba\square$  a ser codificada. Observe que a sequência binária 100111101 correspondente a um intervalo completamente dentro do intervalo associado à sequência  $bbba\square$ , codificando assim a sequência da fonte.

A Figura 40 apresenta todo o processo de subdivisão dos intervalos. Note que, o processo de divisão do intervalo  $[0, 1)$  proporcionalmente às probabilidades dos símbolos da fonte e a subdivisão binária desse intervalo para identificar uma sequência binária que codifique a mensagem podem ocorrer de forma concomitante. À medida que cada símbolo reduz o intervalo com base em sua probabilidade, o algoritmo pode simultaneamente refinar uma representação binária, ajustando-a para permanecer dentro do intervalo atual, o que otimiza o processo de codificação.

A codificação aritmética comprime uma sequência de símbolos em uma única sequência binária por meio da divisão iterativa de um intervalo inicial  $[0, 1)$ . Para cada símbolo  $x_n$  da sequência, o algoritmo utiliza um modelo probabilístico que fornece a distribuição preditiva  $p(x_n = a_i|x_1, \dots, x_{n-1})$ , definindo as probabilidades cumulativas inferiores e superiores:

$$Q_n(a_i|x_1, \dots, x_{n-1}) = \sum_{a' < a_i} P(x_n = a'|x_1, \dots, x_{n-1}), \quad (337)$$

$$R_n(a_i|x_1, \dots, x_{n-1}) = \sum_{a' \leq a_i} P(x_n = a'|x_1, \dots, x_{n-1}). \quad (338)$$

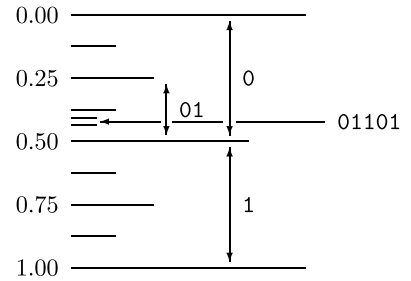


Figura 38: Definição dos intervalos probabilísticos para uma sequência binária na saída do codificador.

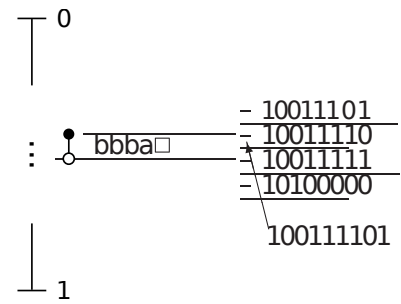


Figura 39: Intervalo final para a sequência de símbolos da fonte e para a sequência binária na saída do codificador.



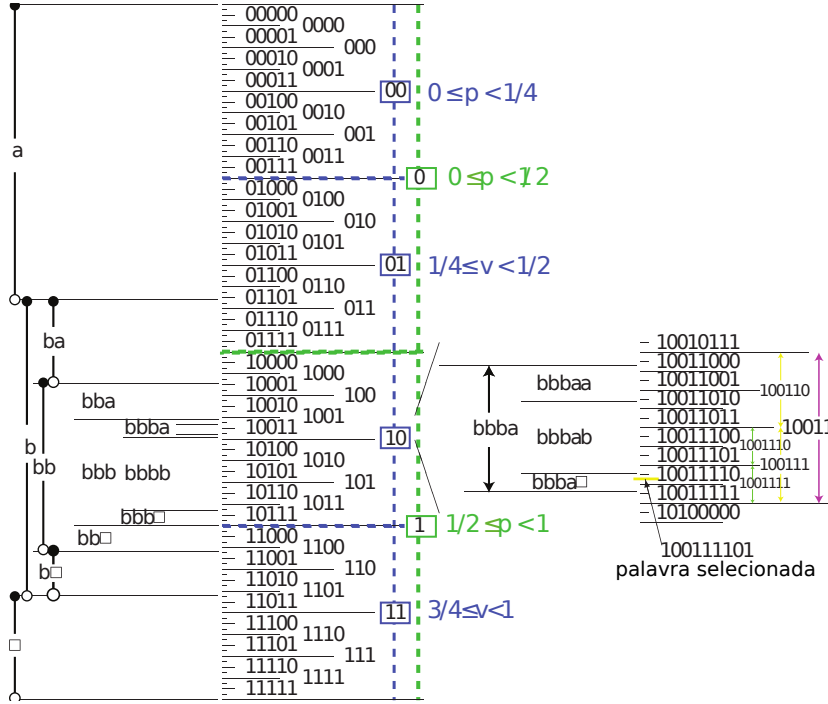


Figura 40: Todo o processo de subdivisão dos intervalos para a sequência de símbolos da fonte e para a sequência binária na saída do codificador (MacKay 2003).

O intervalo atual  $[u, v)$  é subdividido proporcionalmente a essas probabilidades, e o subintervalo correspondente a  $x_n$ , dado por  $[u + (v - u)Q_n, u + (v - u)R_n)$ , é selecionado. Esse processo estreita o intervalo a cada passo. No fim, uma sequência binária é escolhida dentro do intervalo final, permitindo ao decodificador reconstruir a sequência com o mesmo modelo. Essa técnica maximiza a compressão, aproximando-se da entropia da fonte.

- 
- 1: **Entrada:** Sequência  $x_1, x_2, \dots, x_N$ ; modelo probabilístico  $P$
  - 2: **Saída:** Intervalo  $[u, v)$
  - 3:  $u \leftarrow 0.0$
  - 4:  $v \leftarrow 1.0$
  - 5: **for**  $n = 1$  **até**  $N$  **do**
  - 6:   Obtenha  $P(x_n = a_i | x_1, \dots, x_{n-1})$  do modelo
  - 7:   Calcule  $Q_n(a_i) = \sum_{a' < x_n} P(x_n = a' | x_1, \dots, x_{n-1})$
  - 8:   Calcule  $R_n(a_i) = \sum_{a' \leq x_n} P(x_n = a' | x_1, \dots, x_{n-1})$
  - 9:    $v \leftarrow u + (v - u) \cdot R_n(x_n | x_1, \dots, x_{n-1})$
  - 10:    $u \leftarrow u + (v - u) \cdot Q_n(x_n | x_1, \dots, x_{n-1})$
  - 11: **Retorne**  $[u, v)$
- 

Algorithm 1: Codificação Aritmética

Witten, Neal e Cleary (1987) eliminaram a necessidade de cálculos com números reais de precisão arbitrária e operações de ponto flutu-

ante. Eles propuseram uma implementação baseada em aritmética inteira de precisão finita, representando o intervalo de probabilidade ( $[low, high]$ ) com inteiros e escalonando-o dinamicamente para evitar *overflow*, emitindo bits incrementalmente durante o processo. Essa abordagem simplificou os cálculos, reduziu a demanda por recursos de hardware e tornou o algoritmo eficiente o suficiente para aplicações reais de compressão de dados, como em arquivos de texto e imagens, pavimentando o caminho para sua adoção em sistemas práticos de compressão.

### *Codificação Lempel–Ziv*

A codificação Lempel–Ziv, desenvolvida por Ziv e Lempel (1977), marca uma transição significativa na compressão de dados ao abandonar abordagens baseadas exclusivamente em probabilidades estatísticas, como Huffman e a codificação aritmética, em favor de uma estratégia que explora padrões repetitivos em sequências. O algoritmo LZ77, o primeiro da família Lempel–Ziv, estabeleceu as bases para métodos de compressão baseados em dicionários dinâmicos, codificando sequências por meio de referências a trechos já observados no texto. Ao utilizar uma janela deslizante para identificar repetições, o LZ77 substitui trechos redundantes por ponteiros (comprimento e distância).

O algoritmo LZ77 utiliza uma janela deslizante para identificar e substituir sequências repetitivas em uma entrada de dados, dividindo-a em dois componentes: o *search buffer*, com algumas centenas de bytes, que armazena símbolos já processados, e o *look-ahead buffer*, com dezenas de bytes, que contém os próximos símbolos a serem codificados (veja a Figura 41). O codificador examina o *search buffer* (da direita para a esquerda) procurando a maior sequência de símbolos que coincida com o início do *look-ahead buffer*. Ele continua a busca em todo o histórico disponível no *search buffer*, selecionando a correspondência mais longa encontrada ou, em caso de empate, a mais recente para simplificar a implementação. Cada correspondência é codificada como um par (distância, comprimento), indicando quantos bytes retroceder e quantos símbolos coincidem, enquanto símbolos sem correspondência são emitidos diretamente. Após cada codificação, a janela desliza, incorporando novos símbolos ao *search buffer* e avançando no *look-ahead buffer*, repetindo o processo até o fim da entrada.

A Figura 42 apresenta uma etapa da codificação da sequência **sir sid eastman easily teases sea sick seals**. Note como o codificador encontra no passado duas instâncias da sequência **eas**. Decide então gerar um *token* na saída que representa a posição (16) de início desta sequência e o comprimento (3) dela. Por fim, acrescenta ao *token* o último símbolo (**e**) que, ao ser acrescentado na *string* de busca, não

← coded text... sir sid eastman easily teases sea sick seals ... ← text to be read  
search buffer look-ahead-buffer

Figura 41: Janela deslizante no LZ77 (Salomon, Motta e Bryant 2007).

geraria um *match*.

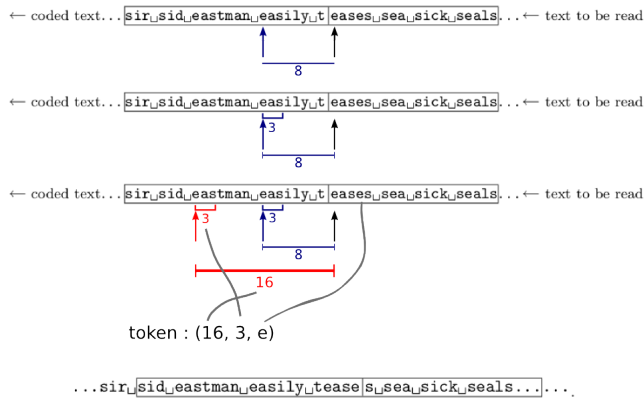


Figura 42: Codificação LZ77 (Salomon, Motta e Bryant 2007).

A escolha da correspondência no LZ77 impacta diretamente a eficiência da compressão. Selecionar a última correspondência encontrada simplifica o codificador, pois elimina a necessidade de armazenar todas as correspondências identificadas durante a busca. No entanto, isso tende a gerar valores de *offset* maiores, aumentando o tamanho da saída codificada e reduzindo a eficiência. Por outro lado, escolher a primeira correspondência resulta em *offsets* menores, o que é mais vantajoso para compressão subsequente. Combinando o LZ77 com a codificação Huffman, onde *offsets* menores recebem palavras de código mais curtas, obtém-se uma abordagem otimizada. Esse método, conhecido como LZH, foi proposto por Bernd Herd, integrando a eficiência do dicionário dinâmico do LZ77 com a compressão estatística do Huffman (Salomon, Motta e Bryant 2007).

Ao longo das décadas de 1980 e 1990, o algoritmo LZ77 foi aprimorado de diversas formas para aumentar sua eficiência e versatilidade na compressão de dados. Entre as principais otimizações, destacam-se: a utilização de campos de *offset* e comprimento com tamanhos variáveis nos *tokens*, permitindo maior flexibilidade na codificação; o aumento do tamanho do *search buffer* e do *look-ahead buffer*, expandindo a capacidade de encontrar correspondências mais longas e distantes; a substituição da janela linear por uma janela circular, que elimina a necessidade de mover o texto inteiro após cada correspondência, reduzindo custos computacionais; e a adição de um bit de *flag* nos *tokens* para distinguir entre literais e referências, eliminando o terceiro campo em alguns casos. Além disso, técnicas como o algoritmo Deflate, empregado no formato ZIP, introduziram tabelas de hash para acelerar a busca por correspondências, combinando LZ77 com codificação Huffman para otimizar ainda mais a compressão. Outras melhorias incluíram o uso de árvores binárias ou estruturas de sufixos para buscas mais rápidas, contribuindo para a adoção do LZ77 em

padrões como GZIP e PNG.

### Codificação LZ78

A codificação LZ78, também conhecida como LZ2, é uma evolução do método LZ77. Diferentemente do LZ77, que utiliza uma janela deslizante com *search buffer* e *look-ahead buffer*, o LZ78 adota um dicionário dinâmico para armazenar strings previamente encontradas, eliminando a dependência de *buffers*. O dicionário, que começa vazio ou com uma string nula na posição zero, cresce à medida que novas strings são processadas, limitado apenas pela memória disponível. O codificador lê os símbolos da entrada, busca correspondências no dicionário e gera *tokens* na forma de pares (ponteiro para o dicionário, símbolo), que representam strings da entrada. Cada *token* adiciona uma nova string ao dicionário, que nunca é esvaziado, podendo crescer rapidamente. Uma estrutura eficiente para o dicionário é uma árvore, com a string vazia como raiz e strings derivadas como filhos, permitindo buscas rápidas mesmo em alfabetos grandes, como os de 256 símbolos (8 bits). Essa abordagem torna o LZ78 particularmente eficaz para compressão de dados com padrões repetitivos, influenciando formatos como LZW e servindo de base para aplicações modernas de compressão.

---

```

1: Entrada: Sequência de símbolos  $S = x_1, x_2, \dots, x_N$ 
2: Saída: Sequência de tokens  $(i, s)$ 
3: Inicialize:
4:    $D \leftarrow$  dicionário vazio com string nula na posição 0
5:    $k \leftarrow 1$  ▷ Próxima posição livre no dicionário
6:    $w \leftarrow$  string vazia ▷ String atual
7:    $T \leftarrow$  lista vazia ▷ Lista de tokens
8: for cada símbolo  $x$  em  $S$  do
9:   if  $w + x$  está em  $D$  then
10:      $w \leftarrow w + x$ 
11:   else
12:      $i \leftarrow$  índice de  $w$  em  $D$ 
13:     Adicione  $(i, x)$  a  $T$ 
14:     Adicione  $w + x$  a  $D$  na posição  $k$ 
15:      $k \leftarrow k + 1$ 
16:      $w \leftarrow$  string vazia
17: if  $w$  não é vazia then
18:    $i \leftarrow$  índice de  $w$  em  $D$ 
19:   Adicione  $(i, \epsilon)$  a  $T$ 
20: Retorne  $T$ 

```

---

Algorithm 2: Codificação LZ78

"sir\_sid\_eastman\_easily\_teases\_sea\_sick\_seals"

Dictionary	Token	Dictionary	Token
0	null		
1	"s" (0, "s")	8	"a" (0, "a")
2	"i" (0, "i")	9	"st" (1, "t")
3	"r" (0, "r")	10	"m" (0, "m")
4	"_j" (0, "_j")	11	"an" (8, "n")
5	"si" (1, "i")	12	"_ea" (7, "a")
6	"d" (0, "d")	13	"sil" (5, "l")
7	"_e" (4, "e")	14	"y" (0, "y")

Figura 43: Exemplo (Salomon, Motta e Bryant 2007).

As Figuras 43 e 44 apresentam a aplicação do LZ78 à string exemplo *sir sid eastman easily teases sea sick seals*. Na Figura 43 é apresentado o dicionário utilizado pelo algoritmo em cada passo da execução, evidenciando o símbolo de entrada em cada etapa, a string de busca e a saída do codificador. A Figura 44 apresenta a árvore em que o dicionário é organizado. Note que as sequências de símbolos são armazenadas como sequências na árvore. Por exemplo, no passo 13 da Figura 43, observamos que a sequência *sil* deverá ser adicionada à árvore. Na Figura 44, podemos observar que 13 (sequência *sil*) foi adicionado abaixo de 5 (sequência *si*), e esta, por sua vez, abaixo de 1 (sequência *s*).

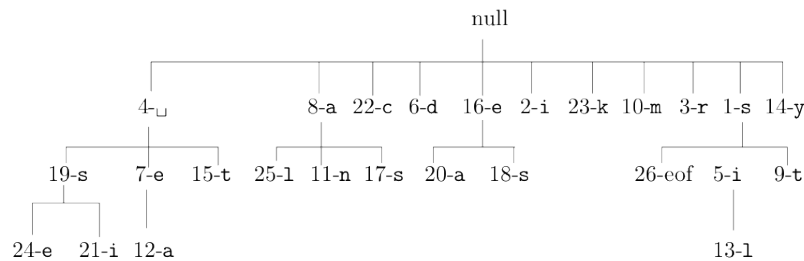


Figura 44: Exemplo (Salomon, Motta e Bryant 2007).

### Codificação LZW

A codificação LZW, desenvolvida por Welch 1984 como uma variação do LZ78, aprimora a abordagem baseada em dicionário ao simplificar a estrutura dos *tokens* e otimizar o uso do alfabeto inicial. Diferentemente do LZ78, que gera *tokens* com dois campos (ponteiro para o dicionário e símbolo), o LZW emite apenas o ponteiro para uma string no dicionário, eliminando o segundo campo e reduzindo a complexidade da saída. O dicionário é inicializado com todas as entradas do alfabeto — por exemplo, 256 símbolos para um alfabeto de 8 bits — permitindo que o codificador comece imediatamente a formar strings sem depender de uma string nula inicial. À medida que processa a entrada, o LZW acumula símbolos em uma string, busca correspondências no dicionário, e adiciona novas strings quando a busca falha, atualizando o dicionário dinamicamente. O LZW é adotado em formatos de compressão como GIF, TIFF e PDF.

O algoritmo LZW, embora eficiente em explorar redundâncias, enfrentou desafios no passado devido ao crescimento do dicionário, que podia esgotar a memória limitada dos computadores da década de 1980. Para contornar isso, implementações práticas frequentemente adotavam limites no tamanho do dicionário ou reinicializações periódicas.

---

```

1: Entrada: Sequência de símbolos  $S = x_1, x_2, \dots, x_N$ ; alfabeto  $A$ 
2: Saída: Sequência de ponteiros  $P$ 
3: Inicialize:
4:    $D \leftarrow$  dicionário com todos os símbolos de  $A$  (e.g., 256 entradas
      para 8 bits)
5:    $k \leftarrow |A| + 1$  ▷ Próxima posição livre no dicionário
6:    $I \leftarrow$  string vazia ▷ String atual
7:    $P \leftarrow$  lista vazia ▷ Lista de ponteiros
8: for cada símbolo  $x$  em  $S$  do
9:   if  $I + x$  está em  $D$  then
10:     $I \leftarrow I + x$ 
11:   else
12:     $i \leftarrow$  índice de  $I$  em  $D$ 
13:    Adicione  $i$  a  $P$ 
14:    Adicione  $I + x$  a  $D$  na posição  $k$ 
15:     $k \leftarrow k + 1$ 
16:     $I \leftarrow x$ 
17: if  $I$  não é vazia then
18:    $i \leftarrow$  índice de  $I$  em  $D$ 
19:   Adicione  $i$  a  $P$ 
20: Retorne  $P$ 

```

---

Algorithm 3: Codificação LZW

Para o exemplo da codificação da sequência `sir sid eastman easily teases sea sick seals`, a Figura 45 apresenta a saída do codificador e a Figura 46 apresenta o dicionário final.

`sir sid eastman easily teases sea sick seals.`

115 (s), 105 (i), 114 (r), 32 (`\`), 256 (si), 100 (d), 32 (`\`), 101 (e), 97 (a), 115 (s), 116 (t), 109 (m), 97 (a), 110 (n), 262 (`\e`), 264 (as), 105 (i), 108 (l), 121 (y), 32 (`\`), 116 (t), 263 (ea), 115 (s), 101 (e), 115 (s), 259 (`\s`), 263 (ea), 259 (`\s`), 105 (i), 99 (c), 107 (k), 280 (`\se`), 97 (a), 108 (l), 115 (s), eof.

Figura 45: Exemplo de saída do codificador LZW (Salomon, Motta e Bryant 2007).

0	NULL	110	n	262	␣e	276	te
1	SOH	...		263	ea	277	eas
...		115	s	264	as	278	se
32	SP	116	t	265	st	279	es
...		...		266	tm	280	s␣
97	a	121	y	267	ma	281	␣se
98	b	...		268	an	282	ea␣
99	c	255	255	269	n␣	283	␣si
100	d	256	si	270	␣ea	284	ic
101	e	257	ir	271	asi	285	ck
...		258	r␣	272	il	286	k␣
107	k	259	␣s	273	ly	287	␣sea
108	l	260	sid	274	y␣	288	al
109	m	261	d␣	275	␣t	289	ls

Figura 46: Dicionário final do LZW para o exemplo dado (Salomon, Motta e Bryant 2007).

## Bibliografia

- Araújo, Leonardo Carneiro, Thaís Cristófar-Silva e Hani Camille Yehia (2013). “Entropy of a Zipfian Distributed Lexicon”. Em: *Glottometrics* 26, pp. 38–49.
- Berger, Roger L. e George Casella (2002). *Statistical Inference*. Duxbury Press.
- Bilmes, Jeffrey A. (2013). *Lecture notes on Information Theory*.
- Cover, Thomas M. e Joy A. Thomas (2006). *Elements of Information Theory*. 2<sup>a</sup> ed. Hoboken, NJ: Wiley-Interscience. ISBN: 978-0-471-24195-9.
- Delfs, Hans e Helmut Knebl (2015). *Introduction to Cryptography: Principles and Applications*. Springer Berlin Heidelberg. ISBN: 9783662479742. DOI: 10.1007/978-3-662-47974-2.
- Ferrer i Cancho, Ramon e Ricard V. Solé (dez. de 2001). “Two Regimes in the Frequency of Words and the Origins of Complex Lexicons: Zipf’s Law Revisited\*”. Em: *Journal of Quantitative Linguistics* 8.3, pp. 165–173. ISSN: 1744-5035. DOI: 10.1076/jqul.8.3.165.4101. URL: <http://dx.doi.org/10.1076/jqul.8.3.165.4101>.
- Gallager, Robert Gray (jan. de 1962). “Low-density parity-check codes”. Em: *IEEE Transactions on Information Theory* 8.1, pp. 21–28. ISSN: 0018-9448. DOI: 10.1109/tit.1962.1057683.
- Golay, Marcel JE (1949). “Notes on digital coding”. Em: *Proc. IEEE* 37, p. 657.
- Hamming, Richard Wesley (abr. de 1950). “Error Detecting and Error Correcting Codes”. Em: *Bell System Technical Journal* 29.2, pp. 147–160. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1950.tb00463.x.
- Hartley, Ralph Vinton Lyon (jul. de 1928). “Transmission of Information”. Em: *Bell System Technical Journal* 7.3, pp. 535–563. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1928.tb01236.x.
- Kneusel, Ronald T. (2018). *Random Numbers and Computers*. Cham: Springer. ISBN: 978-3-319-77696-5. DOI: 10.1007/978-3-319-77697-2.



- Knuth, Donald E. (1997). *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. 3ª ed. Reading, MA: Addison-Wesley. ISBN: 978-0-201-89684-8.
- Kocarev, Ljupco e Shiguo Lian (2011). *Chaos-Based Cryptography: Theory, Algorithms and Applications*. Berlin, Heidelberg: Springer. ISBN: 978-3-642-20541-5. DOI: 10.1007/978-3-642-20542-2.
- MacKay, David John Cameron (2003). *Information theory, inference, and learning algorithms*. Cambridge, UK ; New York: Cambridge University Press. ISBN: 9780521642989.
- Matsumoto, Makoto e Takuji Nishimura (1998). “Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator”. Em: *ACM Transactions on Modeling and Computer Simulation* 8.1, pp. 3–30. DOI: 10.1145/272991.272995.
- Salomon, David, Giovanni Motta e David Bryant (2007). *Data Compression: The Complete Reference*. Springer London. ISBN: 9781846286032.
- Shannon, Claude E. (1948). “A mathematical theory of communication.” Em: *Bell Syst. Tech. J.* 27.3, pp. 379–423.
- (jan. de 1951). “Prediction and Entropy of Printed English”. Em: *Bell System Technical Journal* 30.1, pp. 50–64. DOI: 10.1002/j.1538-7305.1951.tb01366.x.
- Tan, Ying (2016). “GPU-Based Random Number Generators”. Em: *Gpu-Based Parallel Implementation of Swarm Intelligence Algorithms*. Elsevier, pp. 147–165. ISBN: 9780128093627. DOI: 10.1016/b978-0-12-809362-7.50010-8. URL: <http://dx.doi.org/10.1016/B978-0-12-809362-7.50010-8>.
- Welch, Terry Archer (jun. de 1984). “A Technique for High-Performance Data Compression”. Em: *Computer* 17.6, pp. 8–19. DOI: 10.1109/mc.1984.1659158.
- Witten, Ian H., Radford M. Neal e John G. Cleary (jun. de 1987). “Arithmetic Coding for Data Compression”. Em: *Communications of the ACM* 30.6, pp. 520–540. DOI: 10.1145/214762.214771.
- Wolfram, Stephen (2002). *A New Kind of Science*. Champaign, IL: Wolfram Media. ISBN: 978-1-57955-008-0.
- Yeung, Raymond W. (2002). *A First Course in Information Theory*. Springer US. ISBN: 9780306467912.
- Zipf, George Kingsley (1935). *An Introduction to Dynamic Philology*. Cambridge, MA: The MIT Press.
- (1949). *Human Behavior and the Principle of Least Effort*. Cambridge, MA: Addison-Wesley.
- Ziv, Jacob e Abraham Lempel (mai. de 1977). “A universal algorithm for sequential data compression”. Em: *IEEE Transactions on Information Theory* 23.3, pp. 337–343. DOI: 10.1109/tit.1977.1055714.