# Parte I

# Compressão de Fontes com Perdas

# 2
# Fundamentos da Compressão de Fontes

Neste capítulo, são apresentados os conceitos fundamentais da teoria da compressão de fontes com perdas. Esta teoria pode ser dividida em duas partes: a *teoria taxa-distorção* [11], que busca estabelecer os limites teóricos da compressão de fontes, e a *teoria de quantizadores* [18], que tem como objetivo encontrar esquemas práticos de compressão que atinjam estes limites.

Este capítulo está organizado da seguinte forma. A Seção 2.1 apresenta a teoria taxa-distorção; nesta seção, alguns conceitos importantes são definidos e o teorema da compressão de fontes é apresentado. Na Seção 2.2, são apresentados os limites teóricos de compressão para algumas fontes de interesse nesse trabalho. A teoria de quantização é desenvolvida na Seção 2.3, e posteriormente particularizada para quantizadores de taxa fixa e de taxa variável, respectivamente, nas seções 2.4 e 2.5.



## 2.1
## Teoria Taxa-Distorção

Um problema qualquer da teoria taxa-distorção é descrito especificando-se a fonte de informação e a medida de distorção consideradas, dois conceitos que são definidos a seguir.

**Definição 2.1 (Fonte sem Memória)** Uma *fonte de informação sem memória* é uma seqüência semi-infinita de variáveis aleatórias $X_1, X_2, \ldots$ independentes e identicamente distribuídas (i.i.d.) sobre o mesmo conjunto $\mathcal{X}$, o qual é chamado de *alfabeto da fonte.* A fonte é dita *discreta* quando o alfabeto $\mathcal{X}$ é discreto, e *contínua* quando $\mathcal{X}$ é contínuo. Uma vez que $X_1, X_2, \ldots$ são variáveis aleatórias i.i.d., uma fonte sem memória também pode ser denotada por uma variável aleatória genérica $X$.

**Definição 2.2 (Medida de Distorção)** Uma *medida de distorção* é um mapeamento

$$d : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}^+ \tag{2-1}$$

dos pares $(x, \hat{x})$, $x \in \mathcal{X}$, $\hat{x} \in \hat{\mathcal{X}}$ para o conjunto dos números reais não-negativos $\mathbb{R}^+$. A distorção $d(x, \hat{x})$ é uma medida do custo de se representar o símbolo $x$ pelo símbolo $\hat{x}$.

Uma medida de distorção de grande interesse nesse trabalho é a *distorção de erro quadrático*, a qual é dada pela seguinte expressão:

$$d(x, \hat{x}) = (x - \hat{x})^2. \tag{2-2}$$

Estendemos o conceito de distorção para seqüências de $n$ símbolos $x^n = (x_1, \ldots, x_n)$ e $\hat{x}^n = (\hat{x}_1, \ldots, \hat{x}_n)$ através da seguinte definição:

**Definição 2.3 (Distorção entre Seqüências)** A *distorção entre as seqüências* $x^n$ e $\hat{x}^n$ é definida como

$$d(x^n, \hat{x}^n) = \frac{1}{n} \sum_{j=1}^{n} d(x_j, \hat{x}_j). \tag{2-3}$$

Considere agora o problema de representar as $n$ saídas de uma fonte através de um número finito de bits. A seqüência $X^n$ produzida pela fonte é representada através de um índice inteiro $i = \alpha(X^n)$, como mostrado na Fig. 2.1. A partir deste índice, é gerada uma seqüência de reprodução $\hat{X}^n = \beta(i)$, a qual é entregue ao usuário como estimativa de $X^n$.

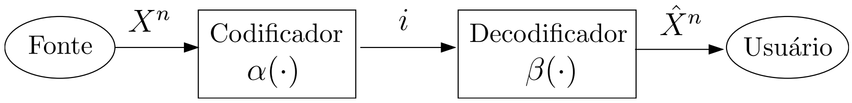


**Figura 2.1:** Código de bloco para compressão.

Como a função $\alpha$ é em geral não-injetora, a seqüência reconstruída pode ser diferente da original, o que significa que o esquema introduz distorção. Deseja-se representar a seqüência $X^n$ através de $\hat{X}^n$ utilizando $nR$ bits e com a menor distorção esperada possível.

**Definição 2.4 (Código de Bloco para Compressão)** Seja o conjunto $\mathcal{I} = \{1, 2, \ldots, M\}$, onde $M = 2^{nR}$. Um *código de bloco para compressão*, de dimensão $n$ e cardinalidade $M$, consiste de uma *função de codificação*

$$\alpha : \mathcal{X}^n \to \mathcal{I} \tag{2-4}$$

que mapeia uma seqüência $x^n \in \mathcal{X}^n$ em um índice $i \in \mathcal{I}$, e uma *função de decodificação (ou reprodução)*

$$\beta : \mathcal{I} \to \hat{\mathcal{X}}^n \tag{2-5}$$

que mapeia um índice $i \in \mathcal{I}$ em uma seqüência $\hat{x}^n \in \hat{\mathcal{X}}^n$.

O código é aplicado a $n$ saídas de uma fonte sem memória de alfabeto $\mathcal{X}$, agrupadas em uma seqüência $X^n \in \mathcal{X}^n$ (Fig. 2.1). O codificador descreve $X^n$ através de um índice $i \in \mathcal{I}$, enquanto o decodificador traduz $i$ em uma seqüência de reprodução $\hat{X}^n \in \hat{\mathcal{X}}^n$. A *distorção* introduzida pelo código é definida como

$$D = E\big[\, d(X^n, \hat{X}^n) \,\big] = E\big[\, d(X^n, \beta(\alpha(X^n))) \,\big]. \tag{2-6}$$

Diz-se que o código tem *taxa* $R$ bits por símbolo da fonte no sentido de que $\log M = nR$ bits são suficientes para representar o índice $i \in \{1, \ldots, M\}$ de cada palavra codificada.

Estamos interessados em construir códigos que sejam bons do ponto de vista do compromisso taxa-distorção, isto é, códigos que, para uma dada taxa $R$, produzem a menor distorção possível, ou inversamente, para uma dada distorção $D$, requerem a menor taxa possível. É conveniente definir uma função que expresse o limite prático dessa busca por códigos, isto é, uma função que expresse a menor taxa $R$ que pode ser atingida por um código em função da distorção tolerada $D$.

**Definição 2.5 (Função Taxa-Distorção Operacional)** A mínima taxa admissível para um código com distorção não superior a $D$ é denotada por $R^*(D)$ e é chamada *função taxa-distorção operacional*. Isto significa que, se uma máxima distorção tolerável $D$ é especificada, então é sempre possível encontrar um código que realize uma compressão a uma taxa superior mas arbitrariamente próxima de $R^*(D)$, enquanto, reciprocamente, é impossível encontrar um código com taxa $R$ inferior a $R^*(D)$. Se todos os pares taxa-distorção $(R, D)$ forem dispostos em um gráfico $R \times D$, então a função taxa-distorção operacional divide o plano em uma região superior admissível (onde códigos com distorção $D$ e taxa $R > R^*(D)$ existem) e uma região inferior em que não existem códigos com distorção $D$ e taxa $R < R^*(D)$.

O problema básico da teoria taxa-distorção é, dada a distribuição da fonte e a medida de distorção, determinar a função $R^*(D)$. O principal resultado dessa teoria, conhecido como Teorema da Compressão de Fontes (Shannon, 1959 [30]), fornece uma expressão matemática para essa função. A seguir, são apresentadas algumas definições que subsidiam o teorema. Por simplicidade, os conceitos apresentados se referem apenas a variáveis aleatórias discretas, mas ressaltamos que existem versões análogas para variáveis aleatórias contínuas [1].

**Definição 2.6 (Entropia)** Seja $X$ uma variável aleatória discreta que assume valores no conjunto $\mathcal{X}$ de acordo com uma distribuição de probabilidade $P(x)$, $x \in \mathcal{X}$. A *entropia* de $X$ é definida como

$$H(X) = -\sum_{x\in\mathcal{X}} P(x) \log P(x). \tag{2-7}$$

**Definição 2.7 (Entropia Condicional e Informação Mútua)** Sejam $X$ e $Y$ variáveis aleatórias discretas com distribuição conjunta $P(x, y)$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$, e distribuições marginais $P(x)$ e $P(y)$, respectivamente. A *entropia condicional* de $X$ dado $Y$ é

$$H(X|Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x, y) \log P(x|y) \tag{2-8}$$

e a *informação mútua* entre $X$ e $Y$ é

$$I(X; Y) = H(X) - H(X|Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x, y) \log \frac{P(x, y)}{P(x)P(y)}. \tag{2-9}$$

**Definição 2.8 (Função Taxa-Distorção)** A *função taxa-distorção* para uma fonte discreta sem memória $X$ com distribuição $P(x)$ e medida de distorção $d(x, \hat{x})$ é definida como

$$R(D) = \min_{P(\hat{x}|x):\sum_{(x,\hat{x})} P(x)P(\hat{x}|x)d(x,\hat{x})\leq D} I(X; \hat{X}) \tag{2-10}$$

onde a minimização é realizada sobre todas as distribuições condicionais $P(\hat{x}|x)$ que satisfazem a restrição no valor esperado da distorção $d(x, \hat{x})$.

**Teorema 2.1 (Teorema da Compressão de Fontes)** A função taxa-distorção operacional para uma fonte discreta sem memória $X$ com distribuição $P(x)$ e medida de distorção $d(x, \hat{x})$ é igual à função taxa-distorção associada. Ou seja,

$$R^*(D) = \min_{P(\hat{x}|x):\sum_{(x,\hat{x})} P(x)P(\hat{x}|x)d(x,\hat{x})\leq D} I(X; \hat{X}) = R(D) \tag{2-11}$$

é a mínima taxa admissível para um código com distorção $D$.

## 2.2
## Cálculo da Função Taxa-Distorção

O cálculo da função $R(D)$ é um problema variacional que envolve uma minimização com restrição de uma função logarítmica. Apesar de conhecida em alguns casos simples, a expressão analítica para a função taxa-distorção é desconhecida na maioria das situações. Na prática, a curva da função $R(D)$ pode ser obtida numericamente utilizando-se o conhecido algoritmo de Blahut [6], o qual é descrito no Apêndice A. Este algoritmo, apesar de definido para fontes discretas, é também capaz de encontrar a função $R(D)$ de uma fonte contínua. Para esse fim, é necessário que a fonte seja discretizada com suficiente precisão [13], isto é, tanto $\mathcal{X}$ quanto $\hat{\mathcal{X}}$ devem ser versões quantizadas dos alfabetos originais.

A seguir são apresentadas curvas da função $R(D)$ para algumas fontes de interesse neste trabalho. Em todos os casos, consideramos a medida de distorção de erro quadrático (Eq. (2-2)).

### Fonte uniforme

Considere uma fonte sem memória com função densidade de probabilidade uniforme dada por

$$f(x) = \begin{cases} 1 & -\frac{1}{2} \leq x \leq \frac{1}{2} \\ 0 & \text{c.c.} \end{cases} \tag{2-12}$$

Esta função é também mostrada na Fig. 2.2.

O gráfico da função taxa-distorção para esta fonte, obtido numericamente com o algoritmo de Blahut, é mostrado na Fig. 2.3.

### Fonte gaussiana

Considere uma fonte sem memória com função densidade de probabilidade gaussiana dada por

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \tag{2-13}$$

onde $\sigma^2$ é a variância da fonte. O gráfico dessa função para $\sigma = 1$ está mostrado na Fig. 2.4.

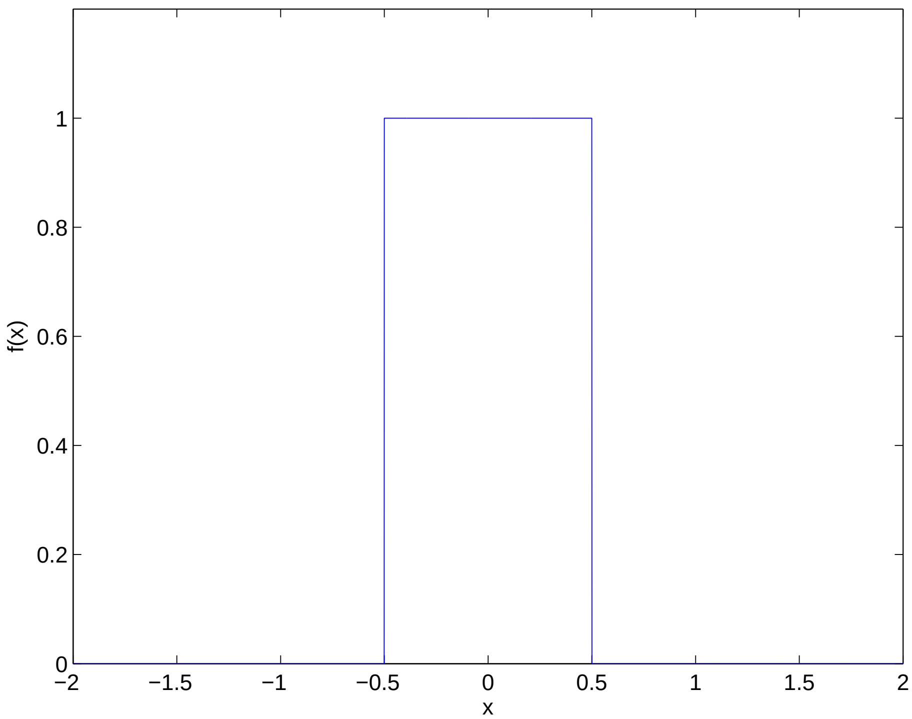


**Figura 2.2:** Função densidade de probabilidade de uma fonte uniforme.

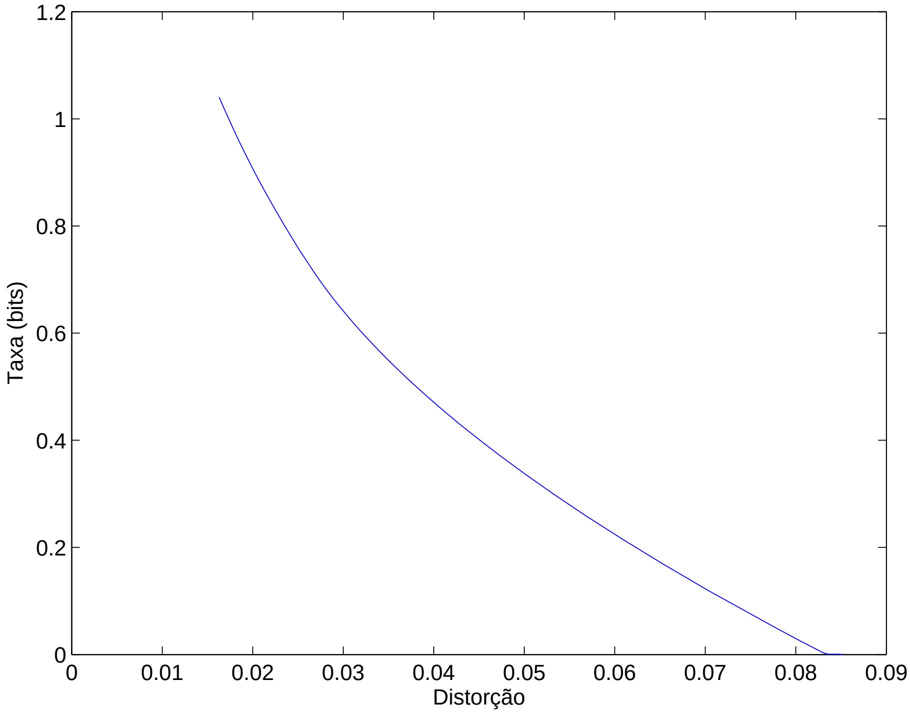


**Figura 2.3:** Função taxa-distorção para uma fonte uniforme.

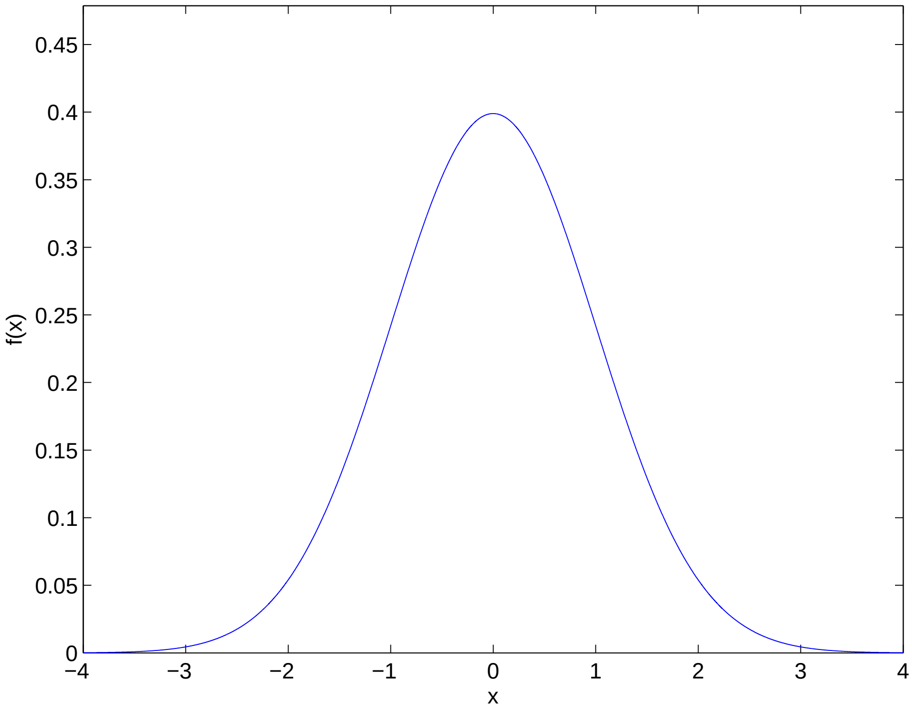


**Figura 2.4:** Função densidade de probabilidade de uma fonte gaussiana.



A função $R(D)$ para esta fonte tem expressão analítica conhecida:

$$R(D) = \begin{cases} \frac{1}{2}\log\frac{\sigma^2}{D} & 0 \leq D \leq \sigma^2 \\ 0 & D \geq \sigma^2. \end{cases} \tag{2-14}$$

Na Fig. 2.5, apresentamos a curva desta função considerando $\sigma = 1$.

### Fonte laplaciana

Considere uma fonte sem memória com função densidade de probabilidade laplaciana dada por

$$f(x) = \frac{1}{2}e^{-|x|}. \tag{2-15}$$

Esta função está esboçada na Fig. 2.6.

O gráfico da função taxa-distorção para esta fonte, obtido numericamente com o algoritmo de Blahut, é mostrado na Fig. 2.7.

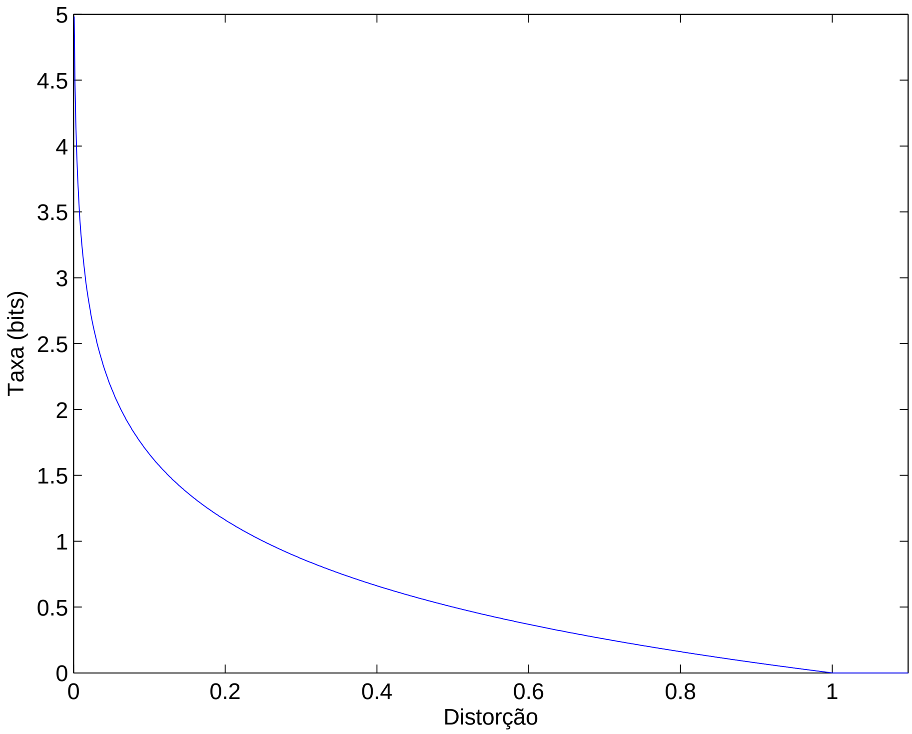


**Figura 2.5:** Função taxa-distorção para uma fonte gaussiana.

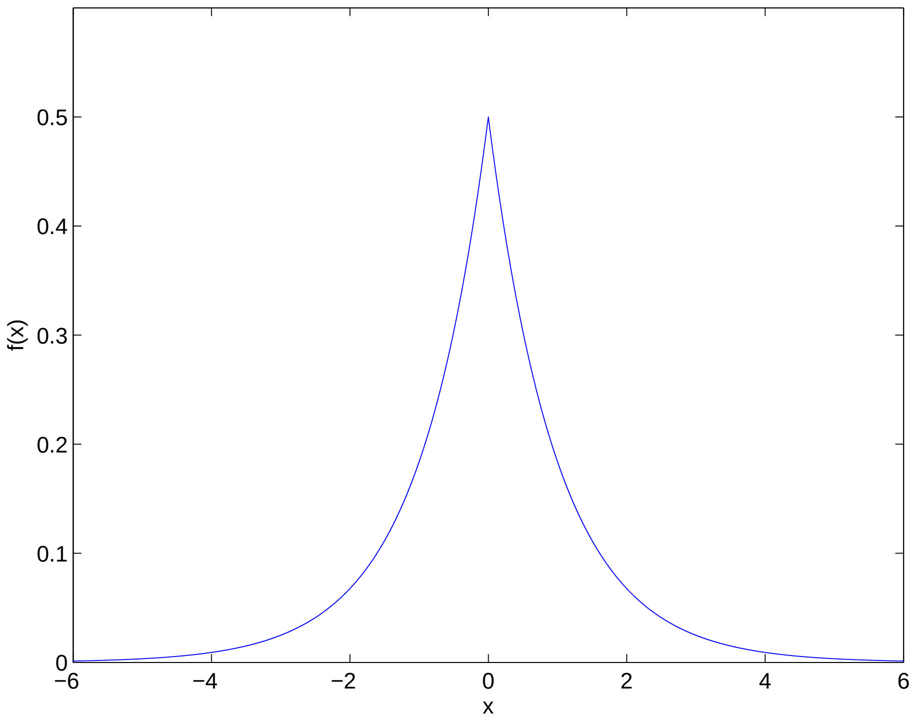


**Figura 2.6:** Função densidade de probabilidade de uma fonte laplaciana.

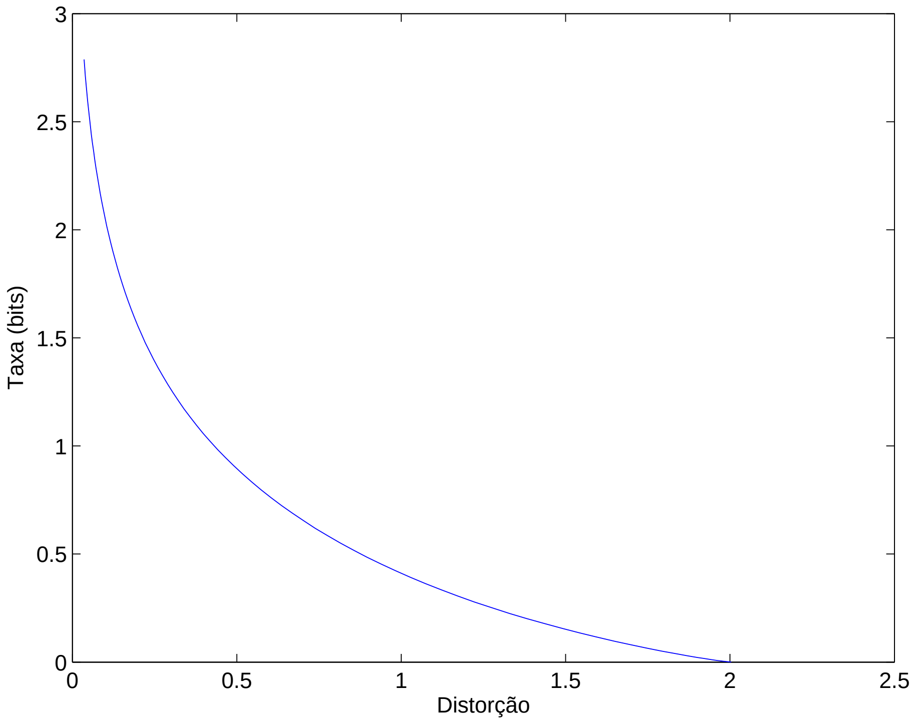


**Figura 2.7:** Função taxa-distorção para uma fonte laplaciana.



## 2.3 Quantização

Um quantizador consiste de um código de bloco para compressão concatenado com um código de fonte sem perdas, onde este último converte os índices produzidos em dígitos binários, com o objetivo de transmití-los através de um canal. O código sem perdas pode ser tanto de taxa fixa como de taxa variável. No primeiro caso, os índices $i$ são convertidos em palavras binárias de comprimento fixo (como implicitamente sugerido pela teoria taxa-distorção), enquanto no segundo caso os comprimentos das palavras binárias podem ser diferentes. A justificativa para essa última abordagem reside na observação prática de que os índices $i$ em geral têm probabilidades significativamente diferentes, o que sugere que uma codificação de taxa variável pode resultar em menor redundância que a codificação de taxa fixa, idealmente atingindo taxas próximas à entropia dos índices. Assim, este tipo de codificação em um quantizador é também chamado de *codificação de entropia*.

Ressaltamos que a função taxa-distorção é também o limitante quando se usa codificação de taxa variável. Assim, se o tamanho de bloco $n \to \infty$, é suficiente considerar a codificação de taxa fixa pois nenhum ganho adicional pode ser obtido com a codificação de taxa variável. Por outro lado, se $n$ é finito, ganhos significativos podem geralmente ser obtidos ao se permitir uma taxa variável na codificação sem perdas.

Usualmente, os quantizadores são utilizados para compressão de uma fonte cujo alfabeto é o conjunto dos números reais. Uma definição de quantizador é apresentada a seguir.

**Definição 2.9 (Quantizador)** Seja o conjunto $\mathcal{I} = \{1, 2, \cdots, M\}$. Um *quantizador* de dimensão $n$ consiste de três componentes (Fig. 2.8): um *codificador com perdas*

$$\alpha : \mathbb{R}^n \to \mathcal{I} \tag{2-16}$$

que mapeia um vetor $n$-dimensional $\boldsymbol{x} = (x_1, \cdots, x_n) \in \mathbb{R}^n$ em um índice $i \in \mathcal{I}$; um *decodificador de reprodução*

$$\beta : \mathcal{I} \to \mathbb{R}^n \tag{2-17}$$

que mapeia um índice $i \in \mathcal{I}$ em um vetor $\hat{\boldsymbol{x}} = (\hat{x}_1, \cdots, \hat{x}_n) \in \mathbb{R}^n$; e um *codificador sem perdas*

$$\gamma : \mathcal{I} \to \mathcal{V} \tag{2-18}$$

que mapeia de forma biunívoca um índice $i \in \mathcal{I}$ em uma palavra binária de um conjunto $\mathcal{V} \subset \{0,1\}^*$, onde $\mathcal{V}$ é um código instantâneo [11] formado por palavras binárias de comprimentos arbitrários.[1]

Alternativamente, um quantizador pode ser especificado por: uma partição de $\mathbb{R}^n$

$$\mathcal{S} = \{\mathcal{S}_i, i \in \mathcal{I}\} \tag{2-19}$$

onde cada $\mathcal{S}_i = \{\boldsymbol{x} : \alpha(\boldsymbol{x}) = i\}$ é uma *região de decisão* ou *célula*; um *dicionário (de reprodução)*

$$\mathcal{C} = \{\boldsymbol{y}_i, i \in \mathcal{I}\} \tag{2-20}$$

onde cada $\boldsymbol{y}_i = \beta(i)$ é uma *palavra de reprodução*; e um dicionário

$$\mathcal{V} = \{v_i, i \in \mathcal{I}\} \tag{2-21}$$

onde cada $v_i = \gamma(i)$ é uma palavra binária.

Se $n = 1$ o quantizador é dito *escalar*; caso contrário, é *vetorial*. Se todas as palavras binárias em $\mathcal{V}$ têm o mesmo comprimento, diz-se que o quantizador tem *taxa fixa*; caso contrário, tem *taxa variável*.

O quantizador é aplicado ao vetor da fonte $\boldsymbol{X} = (X_1, \cdots, X_n) \in \mathbb{R}^n$, o qual é reproduzido na saída como o vetor $\hat{\boldsymbol{X}} = (\hat{X}_1, \cdots, \hat{X}_n)$. A distorção

[1] A notação $\{0,1\}^* = \{0, 1, 00, 01, 10, 11, \cdots\}$ é usada para representar o conjunto de todas as palavras binárias de quaisquer comprimentos.

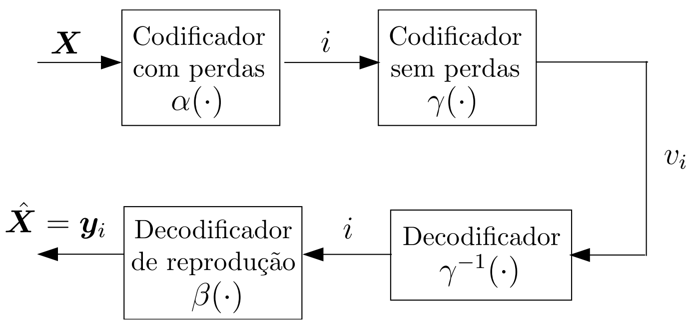


**Figura 2.8:** Quantizador.

associada ao quantizador é definida como

$$D = E\big[\, d(\boldsymbol{X}, \hat{\boldsymbol{X}})\,\big] = E\big[\, d(\boldsymbol{X},\, \beta(\alpha(\boldsymbol{X})))\,\big] \tag{2-22}$$

e a taxa do quantizador, em bits por símbolo da fonte, é definida como

$$R = E[\frac{1}{n}\,|\gamma(\alpha(\boldsymbol{X}))|\,] = \frac{1}{n}\sum_{i=1}^{M} p_i|\gamma(i)| = \frac{1}{n}\sum_{i=1}^{M} p_i\ell_i \tag{2-23}$$

onde $p_i = P(\alpha(\boldsymbol{X}) = i) = P(\boldsymbol{X} \in \mathcal{S}_i)$ é a probabilidade de ocorrência da célula $S_i$, e $\ell_i = |v_i|$ é o comprimento da palavra binária associada.

O projeto de um quantizador tem como objetivo otimizar o compromisso entre taxa e distorção. Este compromisso ótimo entre taxa e distorção pode ser formalizado de três maneiras equivalentes: como uma otimização da taxa dada uma restrição na distorção, como uma otimização da distorção dada uma restrição na taxa, ou ainda como a otimização de um lagrangiano envolvendo taxa e distorção. Nesta última abordagem, a função objetivo que se deseja minimizar é expressa como

$$J_\lambda(\alpha, \beta, v) = D(\alpha, \beta) + \lambda R(\alpha, v) \tag{2-24}$$

onde $\lambda$ é um número real não-negativo. Um baixo valor de $\lambda$ leva a soluções de baixa distorção e alta taxa, enquanto um alto valor de $\lambda$ leva a soluções de baixa taxa e alta distorção.

Não existe solução geral fechada para este problema. É possível, no entanto, encontrar algumas *condições de otimalidade* do quantizador, isto é, condições que precisam ser satisfeitas para que o quantizador seja ótimo. Estas propriedades de um quantizador ótimo são sumarizadas a seguir:

– Para um codificador com perdas $\alpha$ fixo, independentemente do codificador sem perdas $\gamma$, o decodificador de reprodução ótimo $\beta$ é dado pelas palavras de reprodução que minimizam a distorção condicional sobre as respectivas regiões de decisão, isto é, por

$$\beta(i) = \arg\min_{\boldsymbol{y}\in\mathbb{R}^n} E\big[d(\boldsymbol{X},\boldsymbol{y})|\,\alpha(\boldsymbol{X}) = i\big], \quad i = 1,\ldots,M. \tag{2-25}$$

Diz-se que $\beta(i)$ é o *centróide* da célula $\mathcal{S}_i = \{\boldsymbol{x} : \alpha(\boldsymbol{x}) = i\}$. Se a medida de distorção é de erro quadrático, então o centróide de $\mathcal{S}_i$ é simplesmente o valor esperado dos vetores $\boldsymbol{X} \in \mathcal{S}_i$:

$$\beta(i) = E\big[\boldsymbol{X}|\,\alpha(\boldsymbol{X}) = i\big]. \tag{2-26}$$

– Para um codificador com perdas $\alpha$ fixo, independentemente do decodificador de reprodução $\beta$, o codificador sem perdas ótimo $\gamma$ é dado pelo código sem perdas ótimo para a fonte discreta $\alpha(\boldsymbol{X})$, por exemplo, um código de Huffman para esta fonte. Note que $\alpha(\boldsymbol{X})$ é a fonte cujos símbolos são os índices $i$ produzidos pelo codificador com perdas, os quais têm probabilidade de ocorrência dada por

$$p_i \;=\; P(\alpha(\boldsymbol{X}) = i) \;=\; E\big[\,\mathbb{1}(\,\alpha(\boldsymbol{X}) = i\,)\big] \tag{2-27}$$

onde $\mathbb{1}(\cdot)$ é a *função verdade* ou *função indicadora.*[2]

– Para um decodificador de reprodução $\beta$, um codificador sem perdas $\gamma$, e um parâmetro lagrangiano $\lambda$ fixos, o codificador com perdas ótimo é um codificador de mínima distorção (vizinho-mais-próximo) para a medida de distorção lagrangiana modificada

$$\alpha(\boldsymbol{x}) = \arg\min_{i\in\mathcal{I}} \big[\, d(\boldsymbol{x},\beta(i)) + \lambda|\gamma(i)|\,\big]. \tag{2-28}$$

Note que quando o quantizador é restrito a uma taxa fixa, todas as palavras binárias $v_i$ têm o mesmo comprimento e portanto $\gamma$ se reduz simplesmente a uma indexação binária de $\mathcal{I}$. Assim, a segunda propriedade acima se torna irrelevante, e na terceira propriedade a parcela que envolve $\gamma$ pode ser eliminada.

As condições acima nos permitem construir um algoritmo iterativo para o projeto de quantizadores: dado o parâmetro $\lambda$ e um quantizador inicial $(\alpha,\beta,\gamma)$, otimiza-se sequencialmente os componentes $\alpha$, $\beta$ e $\gamma$, considerando para cada um deles os componentes restantes fixos. Este

[2] A *função indicadora* é definida como $\mathbb{1}(B) = \begin{cases} 1 & \text{se a proposição } B \text{ é verdadeira} \\ 0 & \text{se a proposição } B \text{ é falsa.} \end{cases}$

processo então é repetido, reduzindo o lagrangiano (2-24) a cada etapa, até que seja atingido um ótimo local ou global.

O algoritmo descrito acima foi proposto por Lloyd para quantização escalar de taxa fixa em 1957 [26], e estendido para vetores e popularizado por Linde, Buzo e Gray em 1980 [25], sendo conhecido como *algoritmo de Lloyd generalizado* ou *algoritmo LBG*. A formulação lagrangiana do algoritmo foi primeiro proposta por Berger em 1972 [3] e posteriormente generalizada para vetores por Chou *et al.* em 1989 [9].

Note que a aplicação do algoritmo exige o cálculo dos valores esperados em (2-26) e (2-27). A não ser para dimensões $n$ pequenas, esta operação se torna extremamente complexa e dispendiosa do ponto de vista computacional pois requer que seja realizada um integração em $n$ dimensões. A abordagem mais comum para esse problema é substituir o valor esperado por uma média amostral, isto é, o valor esperado de uma variável aleatória é estimado como a média aritmética de uma seqüência de amostras independentes dessa variável. Como todas as variáveis de interesse são funções do vetor aleatório $\boldsymbol{X}$, é suficiente dispor de uma seqüência de amostras desse vetor (ou seja, da fonte). Esta seqüência, quando utilizada no projeto de quantizadores, é geralmente chamada de *seqüência de treinamento*; quando usada na avaliação de desempenho, para o cálculo da distorção em (2-22), é chamada de *seqüência de teste*. Note que esta abordagem é particularmente útil no caso em que a distribuição da fonte é desconhecida, por exemplo, na compressão de dados reais. A justificativa para a substituição do valor esperado por uma estimativa através de seqüências amostrais é melhor desenvolvida no Apêndice B.



## 2.4 Quantização de Taxa Fixa

Nesta seção particularizamos o desenvolvimento da seção anterior para quantizadores de taxa fixa. Como apontado anteriormente, em quantizadores de taxa fixa o codificador sem perdas $\gamma$ se torna trivial. Assim, dados a dimensão $n$ e o número de palavras de reprodução $M$, um quantizador de taxa fixa com codificação ótima pode ser completamente especificado através de seu dicionário de reprodução $\mathcal{C} = \{\boldsymbol{y}_i\}_{i=1}^{M}$, enquanto a regra de codificação ótima (mínima distorção) é dada simplesmente por

$$\alpha(\boldsymbol{x}) = \arg\min_{i \in \mathcal{I}} d(\boldsymbol{x}, \boldsymbol{y}_i). \tag{2-29}$$

Se $M$ é uma potência de 2, é sempre possível codificar sem perdas os $M$ índices $i \in \mathcal{I}$ através de um código binário de comprimento fixo $\ell_i = \log M$. Se $M$ não é uma potência de 2, a indexação requer $\ell_i = \lceil \log M \rceil$ bits. Independentemente de $M$, no entanto, definimos a taxa como

$$R = \frac{1}{n} \log M. \tag{2-30}$$

Existem várias justificativas para se usar essa definição ao invés da definição mais rigorosa $R' = n^{-1} \lceil \log M \rceil$. Em primeiro lugar, a taxa $R$ pode ser alcançada codificando-se conjuntamente um grupo de 2 ou mais índices consecutivos [17]. Para $n$ grande, mesmo esse procedimento é desnecessário pois a diferença $R' - R < 1/n$ torna-se desprezível. Além disso, estamos interessados em analisar o desempenho teórico de quantizadores, e a definição em (2-30) mostra-se mais adequada à análise.

Um quantizador de taxa fixa com medida de distorção de erro quadrático é conhecido como *quantizador de mínimo erro médio quadrático* (MMSE), ou *quantizador Lloyd-Max* no caso escalar. O algoritmo de projeto para esse tipo de quantizador é explicitado a seguir.

**Algoritmo 2.1 (Projeto de Quantizador MMSE)**

1) Especifique a dimensão $n$, o tamanho do dicionário $M$ e o dicionário inicial $\{\boldsymbol{y}_i\}_{i=1}^{M}$.
2) Defina $\alpha(\boldsymbol{x}) = \arg\min_i d(\boldsymbol{x}, \boldsymbol{y}_i)$.
3) Calcule $\boldsymbol{y}_i = E\big[\boldsymbol{X} \,|\, \alpha(\boldsymbol{X}) = i\big], \quad i = 1, \ldots, M$.
4) Volte para o passo 2 até que $\{\boldsymbol{y}_i\}_{i=1}^{M}$ convirja.
5) Fim. ■

## 2.5
## Quantização com Restrição de Entropia

Nesta seção retornamos o desenvolvimento da Seção 2.3 para quantizadores sem a restrição de taxa fixa. Dados a dimensão $n$ e o número de palavras de reprodução $M$, um quantizador genérico com codificação ótima pode ser completamente especificado através do dicionário de reprodução $\mathcal{C} = \{\boldsymbol{y}_i\}_{i=1}^{M}$, das palavras binárias $v_i$, $i = 1, \ldots, M$, e da regra de codificação ótima

$$\alpha(\boldsymbol{x}) = \arg\min_{i \in \mathcal{I}} \big[\, d(\boldsymbol{x}, \boldsymbol{y}_i) + \lambda |v_i| \,\big]. \tag{2-31}$$

Como a restrição é imposta aos comprimentos das palavras binárias $v_i$, os quais normalmente estão ligados às probabilidades dos índices $i$ e conseqüentemente à sua entropia, este tipo de quantizador é chamado de *quantizador com restrição de entropia*, sendo denotado por ECVQ (*entropy-constrained vector quantizer*) no caso vetorial e por ECSQ (*entropy-constrained scalar quantizer*) no caso escalar.

Para fins de análise e projeto, é útil considerar que se dispõe de um codificador de entropia "ideal", isto é, formado por palavras $v_i$ de comprimentos fracionários

$$|v_i| = \log(1/p_i) \tag{2-32}$$

onde $p_i = P(\alpha(\boldsymbol{X}) = i)$. Nesse caso, a "taxa" do quantizador resultante[3] é exatamente igual à entropia da fonte $\alpha(\boldsymbol{X})$:

$$R = \sum_{i=1}^{M} p_i \log(1/p_i) = H(\alpha(\boldsymbol{X})). \tag{2-33}$$

Essa abordagem tem a vantagem adicional de não atar o quantizador obtido a um codificador de entropia específico.

O algoritmo para projeto de um ECVQ com medida de distorção de erro quadrático é explicitado a seguir.

**Algoritmo 2.2 (Projeto de ECVQ)**

1) Especifique a dimensão $n$, o tamanho do dicionário $M$, o dicionário inicial $\{\boldsymbol{y}_i\}_{i=1}^M$ e o parâmetro lagrangiano $\lambda$.
2) Inicialize $p_i = 1/M$, $i = 1 \dots M$.
3) Defina $\alpha(\boldsymbol{x}) = \arg\min_i \left[\, d(\boldsymbol{x}, \boldsymbol{y}_i) + \lambda \log(1/p_i) \,\right]$.
4) Calcule $\boldsymbol{y}_i = E\big[\boldsymbol{X} \,|\, \alpha(\boldsymbol{X}) = i\big], \quad i = 1, \dots, M$.
5) Calcule $p_i = E\big[\, \mathbb{1}(\, \alpha(\boldsymbol{X}) = i \,) \,\big], \quad i = 1, \dots, M$.
6) Volte para o passo 3 até que $\{\boldsymbol{y}_i\}_{i=1}^M$ convirja.
7) Fim. ■

Dependendo do valor de $\lambda$ escolhido, é possível que em alguma iteração do algoritmo nenhum vetor da fonte seja associado a uma certa região

[3] Note que a taxa R' de um código de Huffman satisfaz a cota [9]

$$R \leq R' \leq R + \max_i p_i + 0.086.$$

de decisão. Descarta-se então a palavra de reprodução correspondente, reduzindo-se de uma unidade o parâmetro $M$, e as iterações continuam. Na operação geral do algoritmo, inicia-se com o menor $\lambda$ de interesse (ex: $\lambda = 0$) e o maior $M$ possível dentro das limitações de complexidade. A cada projeto finalizado, $\lambda$ é gradativamente aumentado, acarretando eventualmente a redução de $M$, e o quantizador obtido é utilizado como ponto de partida do projeto seguinte. Quando $\lambda = 0$, o algoritmo se reduz ao do projeto de quantizadores de mínima distorção; quando $\lambda \to \infty$ ($M = 1$), o quantizador resultante é trivial e tem taxa $R = 0$. Este procedimento nos permite traçar uma curva de operação $R \times D$ para um ECVQ semelhante às curvas mostradas nas figuras 2.3, 2.5 e 2.7.

Apesar de mais poderosos que os de taxa fixa, os quantizadores de taxa variável possuem a séria desvantagem prática de precisar lidar com o número variável de bits produzidos [18]. Por exemplo, para se transmitir esses bits através de um canal de comunicação de taxa fixa, é preciso utilizar um buffer e lidar com os problemas de *overflow* e *underflow* de buffer que podem ocorrer. Outra desvantagem é o potencial para propagação de erros quando alguns bits são recebidos com erro (canal não-ideal). Os problemas causados pela variabilidade da taxa podem ser proibitivos para muitas aplicações, exigindo o uso de estratégias complexas para minimizar seus efeitos. Este motivo, combinado ao fato de que a codificação de entropia é geralmente uma etapa significativamente complexa, justifica a busca de quantizadores de taxa fixa com maior dimensão e menor complexidade total.

Os quantizadores apresentados neste capítulo são conhecidos como quantizadores *não-estruturados*, isto é, permite-se que as palavras de reprodução sejam escolhidas com total liberdade em $\mathbb{R}^n$. Uma outra classe de interesse é a de quantizadores *estruturados*, onde é imposta uma estrutura em algum componente do quantizador, por exemplo, na geração do dicionário. Esta abordagem resulta geralmente em códigos sub-ótimos para sua dimensão, porém com implementação mais eficiente ou com outras vantagens práticas. Um exemplo de quantizador vetorial estruturado é apresentado no próximo capítulo.