Trabalho de Conclusão de Curso - Bacharelado em Matemática

Introdução à Teoria de Códigos Corretores de Erros

Vinícius Marques Santorsula

Universidade Federal do ABC

Título: Introdução à Teoria de Códigos Corretores de Erros

Autor: Vinícius Marques Santorsula
Orientador: Prof. Dr. Nazar Arakelian

Trabalho de conclusão de curso apresentado como requisito parcial para obtenção do título de Bacharel em Matemática pela Universidade Federal do ABC.

Banca Examinadora:

Prof. Dr.
Universidade Federal de ..

Prof. Dr.
Universidade Federal de ..

Santo André, X de Março de 2024.

Sumário

Agradecimento

Agradeço a minha família e aos meus professores.

Resumo

Este trabalho tem como objetivo estudar o conteúdo introdutório da Teoria dos Códigos Corretores de Erros.

O trabalho foi desenvolvido devido à motivação durante a gradução em temas correlatos a álgebra abstrata, bem como sua aplicação em teoria da informação e dados.

A elaboração do trabalho baseou-se na pesquisa em bibliografias e artigos publicados sobre o assunto em seu amplo aspecto histórico.

O enfoque foi dado aos Códigos Lineares e suas aplicações abordando a correção e detecção de erros em uma transmissão.

Palavras Chaves: Informação, Códigos, Erros

Abstract

This work aims to study the introductory content of the Theory of Error Correcting Codes.

The work was developed due to motivation during underraduation in topics related to abstract algebra, as well as its application in information and data theory.

The preparation of the work was based on research in bibliographies and articles published on the subject in its broad historical aspect.

The focus was on Linear Codes and their applications addressing the correction and detection of errors in a transmission.

Keywords: Information,Codes, Errors

1 Introdução

A teoria de códigos corretores de erros iniciou-se com a publicação de "A mathematical theory of communication", em Julho de 1948 por Claude Shannon abordando a Teoria de Informações. Atualmente esta teoria tem várias aplicações nas mais diversas áreas, incluindo inferência estatística, processamento de linguagem natural, criptografia, neurociência computacional, evolução e computação quântica.

Os códigos corretores de erros estão presentes no cotidiano de todos nós, basta debruçarmos sobre a necessidade de transmissão e armazenamento de dados com a possibilidade de geração de erros ou ruídos no processo. Assim, a aplicação desta teoria é extremamente útil quando visamos aumentar a precisão e confiabilidade da informação.

Um exemplo simples e cotidiano que podemos recorrer é o CPF, Cadastro de Pessoas Físicas. O número de CPF é composto por 11 dígitos. Os 9 primeiros são os números base, e os 2 últimos são os chamados dígitos verificadores, que são utilizados para validar se os 9 números base estão corretos. Para verificarmos a validade, peguemos os números da base e multipliquemos por sua posição em ordem decrescente, sendo o primeiro dígito a 10^a posição e o último dígito da base a 2^a. Somamos os resultados das respectivas multiplicações e com o resto de sua divisão por 11, podemos comparar com as seguintes regras:

- Se o resto for 0 ou 1, então o primeiro dígito verificador é igual a 0.
- Se o resto for 2, 3, 4, 5, 6, 7, 8, 9 ou 10, então o primeiro dígito verificador é a diferença entre o número 11 e o resto da divisão por 11.

Para o segundo dígito verificador utiliza-se a mesma lógica e regras do primeiro dígito verificador mas tomando o primeiro dígito da base na 11^a posição e incluindo o primeiro dígito verificador na última posição.

Vamos a um exemplo com o CPF 965.963.620-27.

Multiplicando para validar o primeiro dígito verificador, temos:

$$(9 \cdot 10) + (6 \cdot 9) + (5 \cdot 8) + (9 \cdot 7) + (6 \cdot 6) + (3 \cdot 5) + (6 \cdot 4) + (2 \cdot 3) + (0 \cdot 2) = 328$$

Como $328 \mod 11 = 9$, caímos na segunda regra resultando então ao dígito 2.

Agora para o segundo:

$$(9 \cdot 11) + (6 \cdot 10) + (5 \cdot 9) + (9 \cdot 8) + (6 \cdot 7) + (3 \cdot 6) + (6 \cdot 5) + (2 \cdot 4) + (0 \cdot 3) + (2 \cdot 2) = 378$$

Como $378 \mod 11 = 4$, caímos na segunda regra resultando então ao dígito 7 e portando validando o CPF.

Note que ao acrescentarmos os dígitos verificadores construídos em função da base conseguimos recorrer a uma forma de validar a informação e encontrar possíveis erros, aumentando assim a confiabilidade de todo o processo de cadastro.

Abordaremos neste trabalho uma introdução à teoria envolta nos códigos corretores de erros.

2 Conceitos Iniciais

Definições

Para conseguirmos abordar os códigos corretores de erros necessitamos definir primeiramente onde estes irão se aplicar.

Definição 2.1 *Chamamos de alfabeto um conjunto finito A com q elementos e A^n o conjunto formado pela combinação de seus elementos tendo a maior combinação tamanho n.*

Definição 2.2 *Um código corretor de erros é qualquer subconjunto próprio de A^n com $n \in \mathbb{N}$.*

A exemplo, tome o alfabeto utilizado na língua portuguesa, teremos contando todas as letras com suas respectivas acentuações e o espaço em branco um conjunto finito que podemos denominar como A. A maior palavra da língua portuguesa é *"inconstitucionalissimamente"*, contendo 27 letras, o que nos da o A^{27}.

Note que pela **Definição 2.2** a língua portuguesa é um código corretor de erros pois é um subconjunto próprio de A^{27}.

Podemos exibir agora uma noção de distancia entre palavras de A^n.

Métrica de Hamming

Definição 2.3 (Distância de Hamming) *Sejam $u, v \in A^n$. A distância de Hamming entre u e v, é*

$$d(u,v) = |\{i \ ; \ u_i \neq v_i \ , \ 1 \leq i \leq n\}|$$

Exemplificando, em $\{a,b\}^4$ podemos calcular a Distância de Hamming dos seguintes termos:

$$d(aaaa, aaab) = 1$$

$$d(aaab, aaab) = 0$$

$$d(bbab, aaaa) = 3$$

Proposição 2.4 *A Distância de Hamming definida em A^n é uma métrica.*

Demonstração:
Para a Distância de Hamming ser uma métrica ela deve respeitar as seguintes propriedades:

- **Positividade**: $d(u,v) \geqslant 0$ e caso $d(u,v) = 0 \Rightarrow u = v$.
- **Simetria**: $d(u,v) = d(v,u)$.
- **Desigualdade Triangular**: $d(u,v) \leq d(u,w) + d(w,v)$

Note que pela própria construção da Distância de Hamming a positividade se faz verdadeira e para $d(u,v) = 0$ temos que $|\{i\ ;\ u_i = v_i\ ,\ 1 \leq i \leq n\}|\ \therefore\ u = v$.

Agora, veja que para a simetria, podemos partir da definição de Distância de Hamming:

$$d(u,v) = |\{i\ ;\ u_i \neq v_i\ ,\ 1 \leq i \leq n\}| = |\{i\ ;\ v_i \neq u_i\ ,\ 1 \leq i \leq n\}| = d(v,u)$$

Validado a simetria, nos resta então verificar a desigualdade triangular. Para isso, vamos analisar as possibilidades comparando as i-ésimas coordenadas.

Podemos ter $u_i \neq v_i$ ou $u_i = v_i$.

Caso $u_i = v_i$ a contribuição para $d(u,v)$ será zero, e portanto, menor ou igual à contribuição das i-ésimas coordenadas de $d(u,w) + d(w,v)$.
Caso $u_i \neq v_i$ a contribuição para $d(u,v)$ será 1 para cada i, mas note que como $u_i \neq v_i$ não poderemos ter $v_i = w_i$ e $u_i = w_i$ assim a contribuição para $d(u,w) + d(w,v)$

será ao maior ou igual a 1 para cada i (($v_i = w_i$ *e* $u_i \neq w_i$) ou ($v_i \neq w_i$ *e* $u_i = w_i$) ou ($v_i \neq w_i$ *e* $u_i \neq w_i$)).

Logo, $d(u,v) \leq d(u,w) + d(w,v)$.

Como as três propriedades são válidas a Distância de Hamming é uma métrica e também chamada de Métrica de Hamming. □

Definição 2.5 (Disco e Esfera) *Seja $a \in A^n$ e $t \geqslant 0$, $t \in \mathbb{R}$. Definimos respectivamente Disco e Esfera de centro a e raio t como sendo os conjuntos finitos:*

$$D(a,t) = \{u \in A^n;\ d(u,a) \leq t\},$$

$$S(a,t) = \{u \in A^n;\ d(u,a) = t\}.$$

E $|D(a,t)|$ e $|S(a,t)|$ suas respectivas cardinalidades.

Lema 2.6 *Para todo $a \in A^n$ e todo natural $r > 0$ temos:*

$$|D(a,t)| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i$$

Demonstração:

Partimos da definição de Esfera.

$$S(a,i) = \{u \in A^n;\ d(u,a) = i\}.$$

Para um elemento u pertencer à esfera ele deve ter i coordenadas diferentes de a. Como A^n tem q elementos e podemos repeti-los, temos $(q-1)$ possibilidades para as i diferentes coordenadas totalizando $(q-1)^i$. Assim como podemos ter $\binom{n}{i}$ combinações destes elementos, logo:

$$|S(a,t)| = \binom{n}{i} (q-1)^i$$

Note agora que para $i \neq k$, $S(a,i) \bigcap S(a,k) = \emptyset$ e como $D(a,t) = \{u \in A^n;\ d(u,a) \leq t\}$, temos que:

$$D(a,t) = \bigcup_{i=0}^{r} S(a,i) \Rightarrow$$

$$|D(a,t)| = \bigcup_{i=0}^{r} |S(a,i)| \Rightarrow$$

$$|D(a,t)| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i$$

□

Definição 2.7 *Distância Mínima*

Dado um código C ***(Definição 2.2)*** *define-se distância mínima, a menor das distâncias entres os elementos diferentes contidos em C.*

$$d = \min\{d(u,v)\ ;\ u,v \in C\ ,\ u \neq v\}$$

Lema 2.8 *Seja* $c, c' \in C$ *e* $c \neq c'$. *Tome* $\kappa = \left[\frac{d-1}{2}\right]$, *onde* $[r]$ *representa a parte real de r e d a distância mínima em C, então:*

$$D(c,\kappa) \bigcap D(c',\kappa) = \emptyset$$

Demonstração:

Por contradição, tome um x tal que $x \in D(c,\kappa) \bigcap D(c',\kappa)$. *Como o raio dos discos descritos é* κ, *temos que a distância de x até seus respectivos centros será menor que o raio:*

$$d(x,c) \leqslant \kappa \ \ e \ \ d(x,c') \leqslant \kappa\ ,$$

Como a distância de Hamming é uma métrica, podemos utilizar a desigualdade triângular:

$$d(c,c') \leq d(x,c) + d(x,c') \Rightarrow d(c,c') \leq 2\kappa \Rightarrow d(c,c') \leq d-1 \Rightarrow d(c,c') \leq d,$$

Absurdo, como d é a distância mínima em C, $d(c,c') \geqslant d$.

Logo,

$$D(c,\kappa) \bigcap D(c',\kappa) = \emptyset$$

□

Introduziremos agora algumas definições práticas da aplicação de códigos corretores de erros que utilizaremos e aprimoraremos durante o desenvolvimento do trabalho.

Suponha que uma fonte origem esteja tentando transmitir uma informação a um receptor e para isto utiliza um canal. Na prática este canal pode ser uma radiofrequência, um circuito digital, um canal de microondas, uma fita magnética, uma fibra ótica etc.

Neste processo podem haver interferências, isto é, modificações na informação original quando comparada à recebida.

Definição 2.9 *Transmissão e Erro*
Seja $c_n, r_n \in A^n$ respectivamente uma palavra emitida e uma palavra recebida.

O processo $c_n \xrightarrow{transmissão} r_n$ entre o emissor e receptor é chamado de transmissão.

Seja $c_n = c_1^{'}, ..., c_n^{'}$ e $r_n = r_1^{'}, .., r_n^{'}$ onde $c_1^{'}, ..., c_n^{'}$ e $r_1^{'}, .., r_n^{'} \in A$.
Chamaremos de erro todos os

$$c_i^{'} \neq r_i^{'}, \quad i = 1, ..., n.$$

.

Assim, o total de erros gerados em $c_n \xrightarrow{transmissão} r_n$ será a distância de Hamming $d(c_n, r_n)$.

Veja que a geração de erros durante a transmissão pode transformar uma palavra do alfabeto em outra, limitando a identificação da informação de origem.

Definição 2.10 *Detecção e Correção*
Considere $c \xrightarrow{transmissão} r$ onde $c, r \in A^n$.
Diremos que conseguimos detectar g erros em r, se

$$\forall a \in A^n - \{r\}, \ \ a \notin D(c, g).$$

Diremos que conseguimos corrigir g erros com $d(c, r) = g$ se conseguirmos univocamente relacionar c e r e suas respectivas componentes.

Teorema 2.11 *Um código C pode corrigir até $\kappa = \left[\frac{d-1}{2}\right]$ erros e detectar até $d - 1$ erros.*

Demonstração:
Tome t erros na trasnsmissão da palavra c com $t \leq \kappa$. Após a tramissão, com os t erros teremos a palavra r, Logo:

$$d(c, r) = t \ \Rightarrow \ d(c, r) \leq \kappa$$

Sabemos do Lema anterior que:

$$D(c, \kappa) \bigcap D(c', \kappa) = \emptyset,$$

Como d é a distância mínima do código C e por construção $\kappa < d$, sabemos que qualquer outra palavra do código terá distância maior que κ.

Assim, conseguimos única e exclusivamente relacionar r com c, a palavra original, já que

$$r \in D(c, \kappa),$$

Note que podemos incluir d-1 erros na palavra original e ainda conseguiríamos a relação já que não há intersecção dentre os discos. □

Definição 2.12 *Código Perfeito*

O código $C \subset A^n$ será dito perfeito se:

$$\bigcup_{c \in C} D(c, \kappa) = A^n.$$

3 Códigos Lineares

Introdução

Dentre os códigos corretores de erros a classe mais empregada são os códigos lineares. Eles são uma classe de códigos que utilizam princípios de álgebra linear para detectar e corrigir erros que podem ocorrer durante a transmissão de dados. São baseados na ideia de que um conjunto de dados pode ser representado como um vetor em um espaço vetorial. A codificação é realizada através de uma transformação linear, que mapeia o vetor de dados de entrada para um vetor de código em um espaço vetorial de maior dimensão. A decodificação é então realizada revertendo essa transformação.

A principal vantagem dos códigos corretores de erros lineares é a sua eficiência. Eles permitem a detecção e correção de um grande número de erros com um custo computacional relativamente baixo. Além disso, eles são flexíveis e podem ser adaptados para uma ampla gama de aplicações, desde comunicações via satélite até armazenamento de dados. Em resumo, os códigos corretores de erros lineares são uma ferramenta poderosa para garantir a integridade dos dados em um mundo cada vez mais digital. Eles são um campo de estudo fascinante e continuam a ser um tópico ativo de pesquisa na teoria da informação e nas ciências da computação.

Definições

Para podermos facilitar nossas notações, tomaremos K um corpo finito com q elementos como o anterior alfabeto denotado por A^n. Assim, teremos para cada n natural, um K espaço vetorial K^n de dimensão n.

Definição 3.1 *Código Linear*
Um código $C \subset K^n$ será chamado de código linear se for um subespaço vetorial de K^n.

Note que por ser $C \subset K^n$ ser subespaço vetorial, temos, por definição, que C também é espaço vetorial sobre o mesmo corpo de K^n, ganhamos então uma forma de notação dos elementos de C em função de suas bases.

Seja k a dimensão do código C e $v_1, v_2, ..., v_k$ uma de suas bases, podemos denotar os elementos de C como:

$$\lambda_1 v_1 + \lambda_1 v_2 + ... + \lambda_k v_k, \qquad \lambda_i, i = 1, ..., k \in K$$

Definição 3.2 *Dado $x \in K^n$, define-se o peso de x como sendo o número inteiro*

$$\omega(x) := |\{i, \quad x_i \neq 0\}|$$

Ou seja, a distância de Hamming de x até 0.

$$\omega(x) = d(x, 0).$$

Definição 3.3 *O peso de um código linear C é o inteiro $\omega(C)$, dado:*

$$\omega(C) := min\{\omega(x); \quad \forall x \in C - \{0\}\}.$$

Proposição 3.4 *Seja $C \subset K^n$ um código linear com distância mínima d. Temos então:*

1. *$\forall x, y \in K^n$ temos que: $d(x, y) = \omega(x - y)$*
2. *$d = \omega(C)$*

Demonstração:

1. Pela definição, sabemos que:

$$\omega(x-y) = d(x-y, 0) = \Big|\{i \ ; \ x_i - y_i \neq 0 \ , \ 1 \leq i \leq n\}\Big| = \Big|\{i \ ; \ x_i \neq y_i \ , \ 1 \leq i \leq n\}\Big| = d(x, y).$$

2. Note que como C é um subespaço vetorial, este, se municia do fechamento da adição, logo, $\forall x, y \in C, com \quad x \neq y \quad \Rightarrow x - y \in C$ Assim como:

$$d = \min\{d(x, y) \ ; \ x, y \in C \ , \ u \neq v\} \quad e \quad \omega(C) := min\{\omega(x); \quad \forall x \in C - \{0\}\}$$

Pelo item 1., temos que

$$d = \omega(C).$$

□

Note que caso queiramos calcular a distância mínima de um código linear grande, computacionalmente teremos uma demanda impraticável.

A fim de otimizar essa busca, podemos representar os subespaços vetoriais C de um espaço vetorial K^n de duas formas.

- **Imagem:**

  Tome uma base de C, $v_1, v_2, ..., v_k$ e a seguinte aplicação linear:

  $$T: \quad K^k \rightarrow K^n$$

  $$x = (x_1, x_2, ..., x_k) \mapsto v_1x_1 + v_2x_2 + ... + v_kx_k$$

  Como T é uma transformação linear injetora, temos $C = Im(T)$. Assim, ao fazer tal transformação, conseguimos simplificar a grandeza do problema e encontrar os elementos de C. Porém note que por $C \subset K^n$,teremos elementos em K^n que não pertencerão a C sendo necessário o teste (retorna-se o cálculo através da própria transformação $v_1x_1 + v_2x_2 + ... + v_kx_k$).

- **Núcleo:**

  Tome um subespaço C' de K^n complementar de C, ou seja:

  $$C \oplus C' = K^n$$

  E considere a seguinte aplicação com $Ker(H) = C$:

  $$H: \quad C \oplus C' \rightarrow K^{n-k}$$

  $$u \oplus v \mapsto v$$

  Assim, basta verificar, pela definição de núcleo, se dado $v \in K^n$, $H(v)$ é um vetor nulo do espaço vetorial K^{n-k}, facilitando o processamento computacional.

A fim de sintetizar, considere $F_3 = \{0, 1, 2\}$ um corpo finito e $C \subset F_3^4$, onde C é um código corretor de erros linear de base $v_1 = 1011$ e $v_2 = 0112$.

Como a base é composta por v_1 e v_2, temos que dimensão de C é 2, isto aplicado ao corpo F_3 nos dá $3^2 = 9$ elementos.

Pela definição de base, podemos descrever os elementos de C como:

$$\forall c \in C, \;\; c = x_1 v_1 + x_2 v_2, \;\; onde \;\; x_i \in F_3,$$

Agora, podemos representar o código linear C através do núcleo da seguinte transformação linear:

$$\begin{aligned} H: \quad F_3^4 \quad &\rightarrow \quad F_3^2 \\ (x_1, ..., x_4) \quad &\mapsto \quad (2x_1 + 2x_2 + x_3, 2x_1 + x_2 + x_4) \end{aligned}$$

Veja, que como C é o $Ker(H)$, basta verificar se dados $x_1, x_2, x_3 \; e \; x_4 \in F_3^4$ transformados por H formam um vetor nulo em F_3^2, que este estará no $Ker(H)$ e por construção, será um elemento de C.

Equivalência de Códigos Lineares

Definição 3.5 *Isometria*
Sejam V, W espaços vetoriais sobre K, munidos de produto interno.
Dizemos que $T : V \to W$ é uma isometria se T é linear bijetora e $\langle T_u, T_v \rangle = \langle u, v \rangle$ quaisquer que sejam $u, v \in V$.

Definição 3.6 *Seja K um corpo finito. Os códigos lineares C e C' serão linearmente equivalentes se houver uma isometria linear $T : K^n \to K^n$ tal que $T(C) = C'$.*

Para que possamos definir operações práticas a fim de inferir se dois códigos corretores de erros lineares são equivalentes, tomaremos os seguintes resultados:

Lema 3.7 *Sejam K um corpo, π uma permutação de $\{1, ..., n\}$. Então $T_\pi : K^n \to K^n$ é uma isometria linear.*

Demonstração:
Para ser isometria linear devemos provar que a métrica de Hamming é mantida com a transformação. Assim:

$$d(T_\pi(u), T_\pi(v)) = d(u, v) \quad onde, \ u, v \in K,$$

Avaliando as coordenadas de u e v, bem como, a permutação π, temos:

$$u = (u_1, ...u_n) \quad e \quad v = (v_1, ...v_n)$$

$$T(u) = (u_1^{'}, ...u_n^{'}) \quad e \quad T(v) = (v_1^{'}, ...v_n^{'})$$

Agora, tome a definição da distância de hamming:

$$d(u, v) = |\{i \ ; \ u_i \neq v_i \ , \ 1 \leq i \leq n\}|$$

Como a permutação age da mesma forma e mantem a bijeção entre as componetentes de v e u, respectivamente pela definição acima u_i e v_i a distância de Hamming não se alterará, logo $d(T_\pi(u), T_\pi(v)) = d(u, v)$ e T_π é uma isometria linear. □

Lema 3.8 *Sejam K um corpo, π uma permutação de $\{1, ..., n\}$ e $f_i : K \to K$, $i = 1, ..., n$,*

bijeções.Assim, $T_\pi \circ T^1_{f_1} \circ ... \circ T^n_{f_n}$ *é linear se, e somente se, cada* f_i *for linear. Onde:*

$$T^i_{f_i}: \quad K^n \rightarrow K^n$$

$$(x_1, ..., x_n) \mapsto (x_1, ..., f_i(x_i), ..., x_n)$$

Demonstração:
Tome e um escalar.

- Como $G = T_\pi \circ T^1_{f_1} \circ ... \circ T^n_{f_n}$ é linear:

$$G(ev) = eG(u) \Rightarrow$$

$$(T_\pi \circ T^1_{f_1} \circ ... \circ T^n_{f_n})(ev) = e(T_\pi \circ T^1_{f_1} \circ ... \circ T^n_{f_n})(v) \Rightarrow$$
$$T_\pi(ev) \circ T^1_{f_1}(ev) \circ ... \circ T^n_{f_n}(ev) = e[T_\pi(v) \circ T^1_{f_1}(v) \circ ... \circ T^n_{f_n}(v)]$$

  Assim, termo a termo, como:

$$T^n_{f_n}(ev) = e[T^n_{f_n}(v)]$$

  Pela construção da transformação linear, temos:

$$f_n(ev) = e[f_n(v)].$$

  Portanto f_i é linear.

- Agora, como f_i é linear, por construção:

$$f_n(ev) = e[f_n(v)] \Rightarrow T^n_{f_n} \ linear$$

  Como T_π também é linear e a composição de transformações lineares é linear $G = T_\pi \circ T^1_{f_1} \circ ... \circ T^n_{f_n}$ é linear.

□

Com os lemas anteriores conseguimos avaliar a equivalência linear de dois códigos de forma mais ágil.

Corolário 3.9 *Dois códigos lineares C e C em K^n são linearmente equivalentes se, e somente se, existem uma permutação π de $\{1,...,n\}$ e elementos $c_1,...,c_k \in K - \{0\}$ tais que*

$$C' = \{(c_1 x_{\pi(1)},...,c_n x_{\pi(n)}); \quad (x_1,...,x_n) \in C\}.$$

Na prática dois códigos serão linearmente equivalentes se, e somente se, cada um deles puder ser obtido do outro mediante uma sequência de operações do tipo:

- Multiplicação dos elementos numa dada posição fixa por um escalar não nulo em todas as palavras
- Permutação das posições de todas as palavras do cósdigo, mediante uma permutação fixa dos n elementos.

Matriz Geradora de Códigos Lineares

Definição 3.10 *Parâmetros do Código Linear*

Sejam K um corpo finito com q elementos e $C \subset K^n$ um código linear.

Definiremos a terna de inteiros (n,k,d) como os parâmetros do código linear C, onde k é a dimensão do código C sobre K, d é a distância mínima de C ($d = \omega(C)$, provado anteriormente) e n, o tamanho das palavras do espaço vetorial K^n.

Note que com a terna de parâmetros conseguimos encontrar, de imediato, o número de elementos do código C, exibido anteriormente como q^k.

Definição 3.11 *Matriz Geradora*

Seja $B = \{v_1,...,v_k\}$ uma base ordenada de C. Tome a matriz G cujas linhas são formadas pelas coordenadas dos vetores de B, isto é, $v_i = (v_{i1},...,v_{in})$ onde $i = 1,...,k$, ou seja:

$$G = \begin{pmatrix} v_1 \\ \vdots \\ v_k \end{pmatrix} = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ \vdots & \vdots & & \vdots \\ v_{k1} & v_{k2} & \cdots & v_{kn} \end{pmatrix}$$

Assim, a matriz G é denominada matriz geradora de C associada a base B.

Com as definições anteriores e o resultado de equivalência de códigos, podemos de forma matricial, comparar e gerar códigos lineares de forma mais eficiente.

Considere a seguinte transformação linear aplicada a matriz da definição anterior:

$$\begin{aligned} T: \quad K^k &\rightarrow K^n \\ x &\mapsto xG \end{aligned}$$

Considerando as componentes de $x = (x_1, ..., x_k)$, temos na imagem da transformação:

$$T(x) = xG = x_1 v_1 + ... + x_k v_k,$$

Por construção $B = \{v_1, ..., v_k\}$ é uma base ordenada de C, logo temos:

$$T(x) \subset C.$$

Agora se aplicarmos a transformação a todo alfabeto, temos:

$$T(K^k) = C.$$

Note que a matriz G não está diretamente associada ao código C e sim a sua base B. Como uma base de um espaço vetorial pode ser obtida de outra base através de operações como:

- (L1) Permutação de linhas
- (L2) Multiplicação de uma linha por escalar não nulo
- (L3) Adição de um múltiplo escalar de uma outra linha

Assim podemos gerar matrizes diferentes, porem geradoras de um mesmo, ou equivalente código C.

Veja que podemos agora construir códigos lineares a partir de matrizes geradoras. Para isso precisamos definir uma base D, ou seja, uma matriz com linhas linearmente independentes, e através da seguinte tranformação linear:

$$\begin{aligned} T: \quad K^k &\rightarrow K^n \\ x &\mapsto xD \end{aligned}$$

Obteremos então, através da $Im(T)$, um novo código linear.

Para exemplificar, considere o corpo, $K = F_2$, e seja G matriz geradora:

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Note que esta saftisfaz os critérios para tal, visto que suas linhas são linearmente independentes formando uma base.

Considerando agora a transformação linear que nos trás através de sua imagem um código $C = xG$ em F_2^5 :

$$\begin{aligned} T: \quad F_2^3 &\to F_2^5 \\ x &\mapsto xG \end{aligned}$$

A palavra 101 em F_2^3 é decodificada para F_2^5 como 01010 seguindo T.

Tome agora, arbitrariamente, $p = 10101$ com $p \in F_2^5$.

Para decodificar p, ou seja, encontrar sua correspondente originária $p' \in F_2^3$, podemos usar a transformação T descrita acima resolvendo o sistema:

$$p'G = p \Rightarrow (x_1, x_2, x_3)\begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} = (10101),$$

Expandindo:

$$\begin{cases} x_1 + & x_2 + & x_3 = & 1 \\ & x_2 + & x_3 = & 0 \\ & x_1 + & x_3 = & 1 \\ & x_2 + & x_3 = & 0 \\ & x_1 + & x_3 = & 1 \end{cases} \Rightarrow \begin{cases} x_1 = & 1 \\ x_2 = & 0 \\ x_3 = & 0 \end{cases}$$

Assim, temos $p' \in F_2^3$, sendo $p' = 100$.

Pela própria construção deste exemplo o sistema de equações não foi difícil de se resolver, porém com uma matriz G mais complexa isso pode se tornar mais trabalhoso.

Assim, utilizando do conjunto de operações previamente exibidas (L1), (L2) e (L3), conseguimos uma forma mais prática simplificar estas resoluções.

Veja como facilitamos ainda mais a resolução do exemplo acima, operaremos em G com (L3) :

$$\mathrm{G}=\begin{pmatrix}1&0&1&0&1\\1&1&0&1&0\\1&1&1&1&1\end{pmatrix}\xrightarrow[(L3)]{x_1=x_1+(-1)x_3}\begin{pmatrix}0&-1&0&-1&0\\1&1&0&1&0\\1&1&1&1&1\end{pmatrix}\xrightarrow[(L3)]{x_1=x_1+(1)x_2}$$

$$\begin{pmatrix}1&0&0&0&0\\1&1&0&1&0\\1&1&1&1&1\end{pmatrix}\xrightarrow[(L3)]{x_3=x_3+(-1)x_2}\begin{pmatrix}1&0&0&0&0\\1&1&0&1&0\\0&0&1&0&1\end{pmatrix}\xrightarrow[(L3)]{x_2=x_2+(-1)x_1}$$

$$\begin{pmatrix}1&0&0&0&0\\0&1&0&1&0\\0&0&1&0&1\end{pmatrix}=G'$$

Note que $xG' = (x_1\ x_2\ x_3\ x_2\ x_3)$, assim podemos simplificar ainda mais a matriz para uma G'',ficando apenas com as componentes a serem decodificadas $xG' = (x_1\ x_2\ x_3\ )$:

$$G''=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$$

Logo voltando ao problema inicial onde , $p = 10101$ é agora facilmente decodificado extraindo de forma direta $p' = 101$.

Definição 3.12 *Forma Padrão de Matrizes Geradoras*

Uma matriz geradora G de um código C estará na forma padrão se tivermos;

$$G = (Id_k|A)$$

Onde Id_k é a matriz identidade $k \times k$ e A, uma matriz $k \times (n-k)$.

Note que entramos em um problema, de fato nem todos os códigos terão uma matriz geradora na forma padrão. Seja a matriz geradora G do código C sobre F_2^5:

$$G=\begin{pmatrix}0&0&1&0&1\\0&0&0&1&1\end{pmatrix}$$

Note que as duas componentes iniciais da base, são nulas, logo, com as operações definidas (L1), (L2) e (L3) jamais conseguiremos uma matriz geradora no formato padrão para este exemplo.

Entretanto, caso se adicione permutações, agora entre colunas, podemos facilmente obter a matriz:

$$G' = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Que é uma matriz geradora na forma padrão porém de um código C' equivalente a C.

Podemos então nos munir de mais duas operações a fim de encontrar uma matriz geradora de forma padrão de um código C' equivalente a C.

- (C1) Permutação entre colunas.
- (C2) Multiplicação de uma coluna por um escalar não nulo.

Vale evidenciar que caso se utilize de tais operações, estas se aplicarão em todas as palavras do código C, tendo assim efeito sobre as palavras resultantes em C'.

Teorema 3.13 *Dado um código C, existe um código equivalente C com matriz geradora na forma padrão.*

Demonstração:

Tome G uma matriz geradora de C.

$$G = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ \vdots & \vdots & & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{pmatrix}$$

Utilizando as operações (L1), (L2), (L3) e (C1), encontraremos a forma padrão de G.

Pela definição de matriz geradora, temos os vetores da base de C, estes por definição de base são linearmente independentes, logo sabemos que a primeira linha de G não é nula e portanto podemos utilizar a operação (C1) para garantirmos que $g_{11} \neq 0$.

Como $g_{11} \neq 0 \Rightarrow (g_{11})^{-1} \neq 0$, podemos então utilizar (L2) tomando $(g_{11})^{-1}$ como o escalar referido. Com esta operação temos na posição g_{11}, 1.

Note que agora para deixarmos a primeira coluna de G no formato de Id_k, precisamos zerar as demais componentes. Para isso utilizemos a operação (L2) em seguida a operação (L3) da seguinte forma:

Como o primeiro termo da matriz, através da primeira operação, foi transformado em 1, utilizemos (L2) para multiplicar a primeira linha por $(-1)g_{21}$ e posteriormente, com (L3), somemos à segunda linha, colocando então 0 na posição de g_{21}. Fazendo este processo para todas as k linhas obtemos:

$$\begin{pmatrix} 1 & b_{12} & \cdots & b_{1n} \\ 0 & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & b_{k2} & \cdots & b_{kn} \end{pmatrix}$$

Para a segunda coluna, vamos manter a estratégia.

Por ser uma base, certamente a segunda linha também não será nula, utilizando então (C1), garantimos que o elemento da segunda coluna será não nulo. Multiplicando a segunda linha pelo inverso desse elemento, temos:

$$\begin{pmatrix} 1 & c_{12} & c_{13} & \cdots & c_{1n} \\ 0 & 1 & c_{23} & \cdots & c_{2n} \\ 0 & c_{32} & c_{33} & \cdots & c_{3n} \\ \vdots & \vdots & & \vdots & \\ 0 & c_{k2} & c_{k3} & \cdots & c_{kn} \end{pmatrix}$$

Utilizando novamente as operações (L2) e (L3) para zerar as demais posições da coluna, temos:

$$\begin{pmatrix} 1 & 0 & d_{13} & \cdots & d_{1n} \\ 0 & 1 & d_{23} & \cdots & d_{2n} \\ 0 & 0 & d_{33} & \cdots & d_{3n} \\ \vdots & \vdots & & \vdots & \\ 0 & 0 & d_{k3} & \cdots & d_{kn} \end{pmatrix}$$

Repare que encontramos um método para imputar 1 ou 0 em quaisquer k posições da matriz utilizando apenas as operações definidas. Operando sucessivamente teremos a forma padrão equivalente à matriz geradora G de C, G' de C' tal que:

$$G' = (Id_k|A)$$

□

4 Códigos Duais

Introdução

Os códigos duais são também lineares e intimamente relacionados por ortogonalidade à temática do capítulo anterior.

Para esta abordagem, precisaremos definir algumas novas operações e propriedades que abordaremos a seguir.

Definições

Definição 4.1 *Produto Interno*
Dados $u, v \in K^n$ espaço vetorial sobre K, sendo:

$$u = (u_1, ..., u_n) \quad e \quad v = (v_1, ..., v_n)$$

Define-se o produto interno:

$$\langle u, v \rangle = u_1 v_1 + ... + u_n v_n$$

Proposição 4.2 *O produto interno definido acima possui as seguintes propriedades:*

1. ***Simetria:*** $\langle u, v \rangle = \langle v, u \rangle$
2. ***Positividade:*** $\langle v, v \rangle \geq 0$ *com* $\langle v, v \rangle = 0$ *se, e somente se* $v = (0, ..., 0)$
3. ***Distributividade:*** $\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$
4. ***Homogeneidade:*** $\langle \alpha u, v \rangle = \alpha \langle u, v \rangle$ *onde* $\alpha \in K$.

Demonstração:

1. **Simetria:**$\langle u, v\rangle = \langle v, u\rangle$

   Pela definição, temos:

$$\langle u, v\rangle = u_1v_1 + ... + u_nv_n = v_1u_1 + ... + v_nu_n = \langle v, u\rangle$$

2. **Positividade:**$\langle v, v\rangle \geq 0$ com $\langle v, v\rangle = 0$ se, e somente se $v = (0, ..., 0)$

   Aplicando a definição em $\langle v, v\rangle$:

$$\langle v, v\rangle = v_1v_1 + ... + v_nv_n = v_1^2 + ... + v_n^2 \geq 0 \;\; e \;\; \langle v, v\rangle = 0 \Leftrightarrow v = (0, ..., 0)$$

3. **Distributividade:**$\langle u + v, w\rangle = \langle u, w\rangle + \langle v, w\rangle$

   Expandindo $\langle u + v, w\rangle$ nas coordenadas:

$$\langle u + v, w\rangle = (u_1 + v_1)w_1 + ... + (u_n + v_n)w_n = u_1w_1 + v_1w_1 + ... + u_nw_n + v_nw_n =$$

$$(u_1w_1 + ... + u_nw_n)(v_1w_1 + ... + v_nw_n) = \langle u, w\rangle + \langle v, w\rangle$$

4. **Homogeneidade:** $\langle \alpha u, v\rangle = \alpha \langle u, v\rangle$ onde $\alpha \in K$.

   Expandindo $\langle \alpha u, v\rangle$ nas coordenadas:

$$\langle \alpha u, v\rangle = \alpha u_1v_1 + ... + \alpha u_nv_n = \alpha(u_1v_1 + ... + u_nv_n) = \alpha \langle u, v\rangle$$

   De fato, as quatro propriedades usuais de produto interno são satisfeitas.

□

Definição 4.3 *Seja $C \subset K^n$ um código linear, define-se:*

$$C^{\perp} = \{v \in K^n \,;\; \langle v, u\rangle = 0, \;\; \forall u \in C\}.$$

Lema 4.4 *Seja $C \subset K^n$ um código linear com matriz geradora G, então:*

1. *$C^{\perp}$ é um subespaço vetorial de K^n.*
2. *$x \in C^{\perp} \Leftrightarrow Gx^t = 0$*

Demonstração:

1. Para que $C^{\perp}$ seja subespaço vetorial de K^n, devemos conferir se temos em $C^{\perp}$ o fechamento da operação.

   Para isto utilizemos a propriedade distributiva com $u, v \in C^{\perp}$ e $w \in C$, assim sabemos:

   $$\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$$

   Pela definição de $C^{\perp}$, temos que $\langle u, w \rangle = 0$ e $\langle v, w \rangle = 0$. Logo,

   $$\langle u + v, w \rangle = 0 \quad \Rightarrow \quad u + v \in C$$

2. Expandindo $x \in C^{\perp} \Leftrightarrow Gx^t = 0$,

   $$\begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ \vdots & \vdots & & \vdots \\ v_{k1} & v_{k2} & \cdots & v_{kn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} v_{11}x_1 + \ldots + v_{1n}x_n \\ \vdots \\ v_{k1}x_1 + \ldots + v_{kn}x_n \end{pmatrix}$$

   Como $v_i = (v_{i1}, \ldots, v_{in})$ onde $i = 1, \ldots, k$, é base de C, $v_i \in C$ e pela definição de $x \in C^{\perp}$, temos:

   $$\begin{pmatrix} v_{11}x_1 + \ldots + v_{1n}x_n \\ \vdots \\ v_{k1}x_1 + \ldots + v_{kn}x_n \end{pmatrix} = \begin{pmatrix} \langle v_1, x \rangle \\ \vdots \\ \langle v_k, x \rangle \end{pmatrix} = 0$$

□

Definição 4.5 *Código Dual*

Seja o subsespaço vetorial $C^{\perp} \subset K^n$, ortogonal a C, chamaremos então $C^{\perp}$ de código dual de C.

Matrizes Geradoras de Códigos Duais

Proposição 4.6 *Seja $C \subset K^n$ um código linear com dimensão k com matriz geradora G na forma padrão ($G = Id_k|A$), então:*

1. $dim(C^{\perp}) = n - k$
2. *A matriz geradora de $C^{\perp}$ é $H = (-A^t|Id_{n-k})$*

Demonstração:

1. Sabemos que $x \in C^{\perp} \Rightarrow Gx^t = 0$

   Como G está na forma padrão, expandindo como $G = (Id_k|A)$:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & a_{11} & \cdots & a_{1\,(n-k)} \\ 0 & 1 & \cdots & 0 & a_{21} & \cdots & a_{2\,(n-k)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 & a_{k1} & \cdots & a_{k\,(n-k)} \end{pmatrix} \begin{pmatrix} x_1 \\ x2 \\ \vdots \\ x_n \end{pmatrix} = 0 \Rightarrow$$

$$\begin{pmatrix} x_1 + a_{11}x_{k+1} + \cdots + a_{1\,(n-k)}x_n \\ x_2 + a_{21}x_{k+1} + \cdots + a_{2\,(n-k)}x_n \\ \vdots \\ x_k + a_{k1}x_{k+1} + \cdots + a_{k\,(n-k)}x_n \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} + \begin{pmatrix} a_{11}x_{k+1} + \cdots + a_{1\,(n-k)}x_n \\ a_{21}x_{k+1} + \cdots + a_{2\,(n-k)}x_n \\ \vdots \\ a_{k1}x_{k+1} + \cdots + a_{k\,(n-k)}x_n \end{pmatrix} = 0 \Rightarrow$$

$$\Rightarrow \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} + \begin{pmatrix} a_{11} & \cdots & a_{1\,(n-k)} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{k\,(n-k)} \end{pmatrix} \begin{pmatrix} x_{k+1} \\ \vdots \\ x_n \end{pmatrix} = 0 \Rightarrow$$

$$\Rightarrow \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} = \text{-A} \begin{pmatrix} x_{k+1} \\ \vdots \\ x_n \end{pmatrix}$$

   Temos então $x_{k+1}, ..., x_n$ para a base de $C^{\perp}$, assim, $C^{\perp}$ tem q^{n-k} elementos e consenquentemente $dim(C^{\perp}) = n - k$.

2. Note que (Id_{n-k}) garante que $H = (Id_{n-k}|A)$ tenham linhas linearmente independentes (apenas a respectiva componente da diagonal principal não é zerada).

   Tomando então H como uma base, podemos através dela gerar um subespaço vetorial de K^n de dimensão $n-k$.

   Veja que as linhas de H são ortogonais às de G :

$$\text{G}=\begin{pmatrix} 1 & 0 & \cdots & 0 & a_{11} & \cdots & a_{1\,(n-k)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 & a_{k1} & \cdots & a_{k\,(n-k)} \end{pmatrix}, \quad \text{H=-}\begin{pmatrix} a_{11} & \cdots & a_{k1} & 1 & 0 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1\,(n-k)} & \cdots & a_{k\,(n-k)} & 0 & 0 & \cdots & 1 \end{pmatrix}$$

   Analisando as linhas g_i de G e h_j de H, onde $i = 1,...,k$ e $j = 1,...,n-k$, vemos:

$$\langle g_1, h_1 \rangle = 1(-a_{11}) + 0(-a_{21}) + ... + 0(-a_{k1}) + 1a_{11} + 0a_{21} + ... + 0a_{1n-k} = -a_{11} + a_{11} = 0$$

   Repare que para todas as g_i e h_j linhas, teremos :

$$\left\langle g_i, h_j \right\rangle = 0$$

   Sabendo:

$$x \in C^{\perp} \Leftrightarrow Gx^t = 0$$

   Temos que o espaço gerado por H está contido em $C^{\perp}$, mas como este espaço gerado também tem dimensão $n-k$, ele só poderá ser $C^{\perp}$.

   Logo, a matriz geradora de $C^{\perp}$ é $H = (-A^t|Id_{n-k})$.

   □

Equivalência de Códigos Duais

Lema 4.7 *Seja C código linear em K^n. Para toda permutação σ de $1,...,n$, $\forall c \in K^*$ e $\forall j = 1,...,n$, temos que:*

1. $(T_\sigma(C))^\perp = T_\sigma(C^\perp)$
2. $(T_c^j(C))^\perp = T_{c^{-1}}^j(C^\perp)$

Onde:

$$T_\sigma : \quad K^n \rightarrow K^n$$

$$(x_1,...,x_n) \mapsto (x_{\sigma(1)},...,x_{\sigma(n)})$$

$$e$$

$$T_c^j : \quad K^n \rightarrow K^n$$

$$(x_1,...,x_j,...,x_n) \mapsto (x_1,...,cx_j,...,x_n)$$

Demonstração:

1. Tome $(y_1,...,y_n) \in C^\perp$ ortogonal a $(x_1,...,x_n) \in C$, e note que se aplicarmos a σ permutação a $(y_1,...,y_n)$, a ortogonalidade é mantida em relação a $(x_{\sigma(1)},...,x_{\sigma(n)})$, pela própria definição do produto interno, portanto:

   $$\langle x, y \rangle = 0 \Rightarrow \langle x_\sigma, y_\sigma \rangle = 0$$

   Assim, $(T_\sigma(C))^\perp$ será o ortogonal a $T_\sigma(C) = (x_{\sigma(1)},...,x_{\sigma(n)})$ ou seja

   $$T_\sigma(C))^\perp = (y_{\sigma(1)},...,y_{\sigma(n)}) = T_\sigma(y_1,...,y_n) = T_\sigma(C^\perp).$$

2. Tome $(y_1,...,y_n,...,y_n) \in C^\perp$ ortogonal a $(x_1,...,x_j,...,x_n) \in C$.

   O vetor ortogonal à $T_c^j(C)$ será igual ao $C^\perp$ em todas, menos na j-ésima coordenada. Analisemos então sua contribuição ao produto interno $\left\langle T_c^j(C), y \right\rangle = 0$.

   Pela definição: $\left\langle T_c^j(C), y \right\rangle = x_1y_1 + ... + cx_jy_j + ... + x_ny_n$.

Por construção sabemos que $\langle x, y \rangle = 0$, assim para de fato encontrarmos o ortogonal a $T_c^j(C)$, precisamos extinguir a contribuição de $c \in K^*$, portanto temos:

$$((T_c^j(C))^\perp = y_1, ..., \frac{1}{c} y_j, ..., y_n = (T_{c^{-1}}^j(C^\perp)).$$

□

Proposição 4.8 *Sejam C e D dois códigos lineares em K^n. Se C e D são linearmente equivalente, então $C^\perp$ e $D^\perp$ são linearmente equivalentes.*

Demonstração:

Como vimos na seção de Equivalência de Códigos Lineares, existem uma permutação σ de $1, .., n$ elementos e $c_1, ..., c_n \in K^*$ tais que:

$$D = T_\sigma \circ T_{c_1}^1 \circ ... \circ T_{c_n}^n.$$

Utilizando o lema provado anteriormente, temos o resultado direto:

$$D^\perp = (T_\sigma \circ T_{c_1}^1 \circ ... \circ T_{c_n}^n(C))^\perp = T_\sigma \circ T_{c_1^{-1}}^1 \circ ... \circ T_{c_n^{-1}}^n(C^\perp).$$

□

Propriedades entre Matrizes Geradoras Duais e Lineares

Lema 4.9 *Seja C um código de dimensão k em K^n com matriz geradora G. Uma matriz H de ordem $(n-k)\times n$, com coeficientes em K e linhas linearmente independentes é uma matriz geradora de $C^\perp$ se, e somente se:*

$$GH^t = 0$$

Demonstração:
Tome as linhas h_j de H com $j = 1,...,n-k$ e g_i de G com $i = 1,...,k$ e suas respectivas componentes h' de H e g' de G. Podemos expandir a expressão:

$$GH^t = \begin{pmatrix} g'_{11} & \cdots & g'_{1\,n} \\ \vdots & \ddots & \vdots \\ g'_{k1} & \cdots & g'_{kn} \end{pmatrix} \begin{pmatrix} h'_{11} & \cdots & h'_{k1} \\ \vdots & \ddots & \vdots \\ h'_{(n-k)\,1} & \cdots & h'_{(n-k)\,k} \end{pmatrix} = \left(\left\langle g_i, h_j \right\rangle\right)$$

Pela própria definição de $C^\perp$, temos que:

$$\left\langle g_i, h_j \right\rangle = 0 \quad \Rightarrow \quad GH^t = 0$$

□

Corolário 4.10 $(C^\perp)^\perp = C$

Demonstração:
Do lema anterior:

$$GH^t = 0 \quad \Rightarrow (GH^t)^t = (0)^t \quad \Rightarrow (G)^t(H^t)^t = (0)^t \quad \Rightarrow G^tH = 0$$

Logo G é a matriz geradora de $(C^\perp)^\perp$ e portanto,

$$(C^\perp)^\perp = C$$

□

Proposição 4.11 *Seja C um código linear e H uma matriz geradora de $C^\perp$, temos então:*

$$v \in C \quad \Leftrightarrow \quad Hv^t = 0$$

Demonstração:
Como demonstrado anteriormente, sabemos que: $x \in C^{\perp} \quad \Leftrightarrow \quad Gx^t = 0$
Pelo corolário anterior, podemos então:

$$v \in (C^{\perp})^{\perp} \quad \Leftrightarrow \quad Hv^t = 0 \quad \Rightarrow \quad v \in C \quad \Leftrightarrow \quad Hv^t = 0$$

□

Definição 4.12 *A matriz geradora H de $C^{\perp}$ é chamada de matriz teste de paridade de C.*

Note que agora temos uma outra forma de avaliar se os elementos $v \in K^n$ pertencem ao código C.
Anteriormente desenvolvemos métodos e operações a fim de simplificar esta avaliação. Em geral utilizávamos como insumo, G matriz geradora de C e sua definição, recorrendo á propriedade:

$$xG = v$$

Ou seja, avaliávamos se o elemento v poderia ser construído através da base de C, representada por G.
Contudo, caíamos na dificultade de resolução do sistema de n equações com k incógnitas $x = (x_1, ..., x_k)$ e apesar das simplificações de G com o uso das operações (L1), (L2), (L3), (C1) e (C2) visando a forma padrao $G = (Id_k|A)$ e consequentemente a otimização da resolução do sistema, ainda poderíamos ter dificuldades para tal a depender da dimensão de G.

Com o resultado anterior, por uma condição de anulamento, utilizando H, matriz teste de paridade de C podemos delimitar seus próprios elementos dada a seguinte característica:

$$v \in C \quad \Leftrightarrow \quad Hv^t = 0$$

Dada a vantagem trazida por tal propriedade definiremos:

Definição 4.13 *Seja C um código linear com H matriz teste de paridade e $v \in K^n$.*
O vetor Hv^t é chamado então de síndrome de v.

Tome agora o seguinte exemplo:
Seja C um código linear sobre F_2 com a seguinte G matriz geradora.

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Como a matriz G está na forma padrão $G = (Id_k|A)$, fica fácil encontrarmos H através da relação :

$$H = (-A^t|Id_{n-k})$$

Logo,

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Para evidênciar a facilidade na verificação dos elementos de C através da síndrome, tome $v_1 = (100111)$ e $v_2 = (010101)$. Testaremos se $v_1 \ \ e \ \ v_2 \in C$. Assim aplicando á definição de sindrome de v_1 e v_2, temos:

$$Hv_1^t = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = 0 \quad \Rightarrow \quad v_1 \in C$$

$$Hv_2^t = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \neq 0 \quad \Rightarrow \quad v_2 \notin C$$

Vimos que o teste de anulamento se mostra mais prático na verificação de elementos de C. Ainda, a matriz de paridade tráz consigo informações sobre o valor do peso d do código.

Proposição 4.14 *Seja H a matriz teste de paridade de um código C. Temos que o peso de C é maior do que ou igual a s se , e somente se , quaisquer $s-1$ colunas de H são linearmente independentes.*

Demonstração:

- $\Leftarrow$

  Suponha então que $s-1$ colunas de H são linearmente independentes. Tome também, $c = c_1,...,c_n$ tal que $c \neq 0$ um elemento de C. Como $Hc^t = 0$, expandindo temos:

  $$Hc^t = \sum c_i h^i, \qquad \text{onde } h^1,...,h^n \text{ são colunas de } H.$$

  Pela definição de peso, $w(c) \leqslant s-1$, visto que $s-1$ é o número de colunas de H e portanto o número de componentes da base de c.

  Assim, para que $Hc^t = 0$ deveríamos ter uma combinação nula de um número t de colunas, com $1 \leqslant t \leqslant s-1$.

  Como H é uma base entramos em contradição. Logo, $s \leqslant w(c)$ e pela definição de peso de um código linear, temos, $s \leqslant w(C)$.

- $\Rightarrow$

  Suponha agora, que $s \leqslant w(C)$ e por absurdo, tome H com $s-1$ colunas $h^1,...,h^{s-1}$ linearmente independentes.

  Como H forma uma base, teríamos $c = (c_1,...,c_{s-1}) \in C$ tal que:

  $$c_1 h^1 + ... + c_{s-1} h^{s-1} = 0$$

  Assim para satisfazer a equação teriamos que ter componentes nulas em c, consequentemente, $w(c) \leqslant s-1 \leqslant s$, absurdo.

□

Teorema 4.15 *Seja H a matriz teste de paridade de um código C. Temos que o peso de C é igula a s se, e somente se, quaisquer $s-1$ colunas de H são linearmente independentes e existem s colunas linearmente dependentes.*

Demonstração:
Suponha que o peso do código C é igual a s, assim, teremos $s-1$ colunas em H linearmente independentes e s colunas serão dependentes pois caso contrário pela proposição anterior teríamos $w(C) = s+1$.

Reciprocamente, tome o conjunto $s-1$ de colunas de H linearmente independetes e s colunas dependentes. Pela proposição anterior, então temos $w(C) \geqslant s$, mas não

podemos ter $w(C) > s$ pois assim todas as colunas s seriam linearmente independetes e chegaríamos a uma contradição. □

Corolário 4.16 *Cota de Singleton*
Os parâmetros (n,k,d) de um código linear satisfazem:

$$d \leqslant n-k+1.$$

Demonstração: Como o posto de uma matriz teste de paridade é $n-k$, temos do teorema anterior:

$$n-k \geqslant d-1 \quad \Rightarrow \quad d \leqslant n-k+1.$$

□

Este corolário define um particular tipo de código linear que vale menção:

Definição 4.17 *Códigos Lineares MDS*
Um código linear será chamado de MDS (Maximum Distance Separable) se:

$$d = n-k+1$$

5 Decodificação

Introdução

De forma prática, a decodificação dos códigos corretores de erros é o enfoque na aplicação desta teoria, visto ser o procedimento de detecção e correção de erros.

O método de decodificação abordado a seguir é baseado nos estudos de David Slepian. Este matemático junto a nomes consagrados já até mencionados neste trabalho como Claude Shannon autor de "A Mathematical Theory of Communication", em Julho de 1948 e Richard Hamming introdutor do conceito de distância de Hamming em 1950, trabalharam na mesma época no Bells labs, desenvolvendo e aprimorando o que futuramente seria nomeado Teoria da Informação.

Definições

Definição 5.1 *Vetor Erro*
Defini-se o erro e como a diferença entre a palavra recebida r e a palavra transmitida c.

$$e = r - c$$

Note que o erro é um vetor de marcação. Para exemplificar tome a palavra transmitida $c = (010011)$ e a recebida $r = (101011)$, então o erro se dará:

$$e = \ r - c \ = \ (101011) - (010011) \ = (111000)$$

Repare que nas componentes onde ouve divergência na transmissão é marcado no vetor erro com 1 em contra partida às componentes corretas na transmissão que são marcadas com 0.

Sendo assim, $w(e)$ corresponde a quantidade de erros ocorridos na transmissão da palavra c sendo recebida como r.

Dado que estas palavras estarão sob a estrutura de um código, podemos aproveitar o ferramentário desenvolvido anteriormente.

Proposição 5.2 *Seja $c \in C$, tal que C seja um código linear com H matriz de paridade. A síndrome do erro na transmissão de c é igual a síndrome da palavra recebida r.*

Demonstração:
Fazendo a síndrome do erro, temos:

$$He^t = H(r-c)^t = H(r^t - c^t) = Hr^t - Hc^t$$

Como $c \in C \;\Rightarrow\; Hc^t = 0$, logo,

$$He^t = Hr^t$$

□

Lema 5.3 *Seja C um código linear em K^n com capacidade de correção κ. Se $r \in K^n$ e $c \in C$ são tais que $d(c,r) \leqslant \kappa$, então existe um único vetor e com $w(e) \leqslant \kappa$, cuja síndrome é igual a síndrome de r tal que $c = r - e$*

Demonstração:
Na seção anterior esta propriedade sobre os pesos já fora demonstrada, assim, sabemos:

$$w(e) = w(r-c) = d(r,c) \leqslant \kappa$$

Para findar a demonstração necessitamos mostar a unicidade.
Assim, suponha dois erros $e = (\alpha_1, ..., \alpha_n)$ e $e^{'} = (\alpha_1{}', ..., \alpha_n^{'})$, logo, teremos seus respectivos pesos que seguem a propriedade acima, sendo $w(e) \leqslant \kappa$ e $w(e^{'}) \leqslant \kappa$.

Suponha agora que tenham a mesma síndrome que r. Como H é uma matriz paridade de C, temos a seguinte igualdade:

$$H(e)^t = H(e^{'})^t$$

Podemos expandir a matriz H denotando suas colunas porh_i e $i = 1, ..., n$. Assim:

$$H(e)^t = H(e^{'})^t \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i h^i = \sum_{i=1}^{n} \alpha_i^{'} h^i,$$

Veja que para $i = 1, ..., n$ tem-se a dependência linear entre 2κ colunas de H. Como $2\kappa \leqslant d-1$ e como sabemos dos teoremas anteriores que quaisquer $d-1$ colunas de H são linearmente independentes temos que $e' = e$, provando a unicidade. □

Algoritmo de Decodificação

A unicidade do peso do erro nos da um resultado parecido à visão inicial, onde os discos não se intersectavam, assim temos a garantia que caso saibamos o vetor e conseguiremos corrigir a palavra r recebida a fim de encontrar unicamente a c transmitida.

O problema agora se configura na determinação de e através de Hr^t.

Tomemos alguns exemplos:

- Seja C um código linear com distância mínima $d \geqslant 3$ e que o peso do erro da transmissão de $c \xrightarrow{\text{transmissão}} r$, $w(e) \leqslant 1$. Assim note que podemos fazer uma primeira averiguação para, de fato, avaliar se houve erros na transmissão.

  Se $He^t = 0 \ \rightarrow Hr^t = 0$ e logo $r \in C$ sendo $r = c$.

  Suponhamos agora que $He^t \neq 0$, logo por construção do exemplo teremos um erro na transmissão e pela própria definição de vetor erro, teremos apenas uma coordenada não nula, $e = (0,...,\alpha,...,0)$ com $\alpha \neq 0$ na i-ésima coordenada, portanto a síndrome de e se dará por $He^t = \alpha h^i$, onde h^i é a i-ésima coluna de H.

  Note que decaímos no problema do desconhecimento de e, mas da forma que o exemplo fora estruturado, conseguimos supor todas as coordenadas de e e além disso, i é bem determinado pois sabermos que $d \geqslant 3$.

- Considere agora um código C com matriz teste de paridade H determinada:

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

  Seja $r = (10100)$ uma palavra recebida, tal que $c \xrightarrow{\text{transmissão}} r$.

  Sabemos que:

$$He^t = Hr^t = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = h^4 \quad \Rightarrow \quad He^t = h^4$$

Assim, sabemos que há erro na quarta coordenada de r, logo, $e = (00010)$.

Temos então:

$$c = r - e = (10110)$$

Note que nos exemplos seguimos uma linha de raciocínio a fim de encontrar e corrigir os erros da transmissão. Podemos então formalizar um algoritmo para tal:

Algoritmo de Decodificação para um Erro

Seja C um código, H sua matriz de paridade e $c \xrightarrow{\text{transmissão}} r$.

(1.) Calcule Hr^t.

(2.) Se $Hr^t = 0$, não há erros, assuma $r = c$.

(3.) Se $Hr^t = s^t \neq 0$, compare s^t com as colunas de H.

(4.) Se $s^t = \alpha h^i$, com $\alpha \in K$, então e é a n-upla com α na posição i e demais coordenadas nulas. Corrija r através de $c = r - e$.

(5.) Caso **(4.)** não for verdadeiro, então mais de um erro foi cometido.

Visto a limitação deste algoritmo a um erro, traremos a estrutura necessária para ampliar seu âmbito de aplicação.

Definição 5.4 *Classe Lateral*
Seja $C \subset K^n$ um código linear, definimos o conjunto chamado classe lateral de $v \in K^n$ segundo C como:

$$v + C = \{v + c; \quad c \in C\}$$

Note que por C ser subespaço vetorial, temos :

$$v + C = C \iff v \in C$$

Lema 5.5 *Sejam $u, v \in K^n$, $Hu^t = Hv^t$ se, e somente se, $u \in v + C$*

Demonstração:
Como $Hu^t = Hv^t \Leftrightarrow Hu^t - Hv^t = 0 \Leftrightarrow H(u-v)^t = 0 \Leftrightarrow u - v \in C$ Tomando $c = u - v$, pela definição de $v + C$, temos $v + c = v + u - v = u \in v + C$ □

Definição 5.6 *Elemento líder*
Um vetor de peso mínimo numa classe lateral é chamado de elemento líder da classe.

Proposição 5.7 *Unicidade do Elemento líder*
Tome $C \in K^n$ um código linear com d distância mínima e $u \in K^n$.
Seja

$$w(u) \leqslant \left[\frac{d-1}{2}\right] = \kappa.$$

Então u é o único elemento líder de sua classe.

Demonstração:
Para provar a unicidade, tomemos $u, v \in K^n$ com $w(u) \leqslant \kappa$ e $w(v) \leqslant \kappa$, se $u - v \in C$ encontrando seu peso, temos:
Anteriormente vimos:

$$w(u - v) = d(u, v)$$

Como a distância de Hamming é uma métrica, podemos usar a desigualdade triângular e a propriedade de simetria:

$$w(u - v) = d(u, v) \leqslant d(u, 0) + d(0, v) = d(u, 0) + d(v, 0)$$

Pela definição de peso temos:

$$w(u - v) \leqslant w(u) + w(v)$$

Como $w(u) \leqslant \kappa$ e $w(v) \leqslant \kappa$ e pela definição de κ, temos:

$$w(u - v) \leqslant 2\kappa \quad \Rightarrow \quad w(u - v) \leqslant d - 1$$

Logo,

$$u - v = 0 \quad \Rightarrow \quad u = v$$

Assim, o elemento líder é único em sua classe. □

De forma mais abrangente podemos agora definir um algoritmo capaz de decodificar não mais apenas um erro, mas sim até a capacidade de correção máxima, κ, de um código.

Para isso, se faz necessário calcular todos os elementos líderes das classes desse código. Como visto na propriedade anterior, podemos selecionar todos os elementos u tal que $w(u) \leqslant \kappa$ e pela unicidade, teremos que cada u será lider de uma e somente uma classe. Com esta listagem de líderes feita, calcula-se suas respectivas síndromes e organize em uma tabela de insumo para o algoritmo.

Algoritmo de Decodificação para κ erros

Seja C um código, H sua matriz de paridade e $c \xrightarrow{\text{transmissão}} r$.

(1.) Calcule $s^t = Hr^t$.

(2.) Se s está na tabela, seja L o elemento líder da classe determinada por s, troque r por $r - L$

(3.) Se s não está na tabela, foram cometidos erros além da capacidade máxima de correção do código.

Agora note, como $He^t = Hr^t$, temos que a classe lateral de e pode ser mapeada pela síndrome de r.

Se o código tiver capacidade de corrigir a quantidade de erros da transmissão, isto é, $w(e) \leqslant \kappa$, pela unicidade do elemento líder, e será único em sua classe e encontrado dentre os elementos calculados da tabela.

Como temos a relação $c = r - e$ podemos corrigir o vetor r através de $c = r - L$.

Afim de exemplificar o algoritmo, tome o exemplo a seguir:

Considere o código C com matriz teste de paridade H a seguir, e distancia mínima 3.

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Como a $d = 3 \quad \Rightarrow \quad \kappa = 1$.

Assim os elementos líderes u, terão $w(u) \leqslant 1$.

Construindo a tabela com as síndromes:

Líder	Síndrome
000000	000
000001	101
000010	011
000100	110
001000	001
010000	010
100000	100

Utilizando o algoritmo para os seguintes casos:

- $r = (100011)$ Fazendo então o **(1.)** do algoritmo.

  $$s^t = Hr^t = (010)^t$$

  Fazendo o **(2.)** do algoritmo. Temos o líder (010000) e consequentemente:

  $$c = r - (010000) = (100011) - (010000) = (110011)$$

- $r = (111111)$ Fazendo então o **(1.)** do algoritmo.

  $$s^t = Hr^t = (111)^t$$

  Como s^t não está na tabela, caímos no **(3.)** do algoritmo. Consequentemente temos mais erros do que o código é capaz de corrigir.

Referências Bibliográficas

[1] HEFEZ, A., VILLELA, M. L. T.; *Códigos Corretores de Erros - Série de Computação e Matemática*, IMPA, 2002.

[2] MACWILLIAMS, F. J. AND SLOANE, N. J. A.; *The Theory of ErrorCorrecting Codes*, North-Holland, New York, 1996.

[3] HUFFMAN, W. C. AND PLESS, V.; *Fundamentals of Error-Correcting Codes*, Cambrige 2003.

[4] C. E. SHANNON; *A Mathematical Theory of Communication*, Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656,July and October, 1948.

[5] ANTON, H. e RORRES, C.; *Álgebra Linear com Aplicações*, Bookman, 8ª Edição, 2001.

[6] BOLDRINI, J. L., COSTA, S. I. R., FIGUEIREDO, V. L. e WETZLER, H. G.; *Álgebra Linear*, Harbra, 3ª Edição, 1986.