Corrigindo erros por meio de códigos lineares *

Robson Ricardo de Araujo e Antonio Aparecido de Andrade †

Resumo

Desde os trabalhos de Claude Shannon, em 1948, o avanço tecnológico na área das telecomunicações tem sido notável. Um grande problema na transmissão de mensagens por algum canal sempre residiu no fato de que, ao atravessar o canal, o conteúdo transmitido sofre distorções e chega modificado ao destinatário, o que impossibita a sua leitura correta. Graças ao trabalho de Shannon, esse problema obteve uma solução. Para identificar erros na transmissão de uma mensagem e corrigí-los, criaram-se os códigos corretores de erros, dos quais trataremos neste artigo, com destaque à classe dos códigos lineares. Para tanto, serão dados alguns resultados importantes relacionados aos corpos finitos, que são estruturas algébricas importantes sobre as quais se constroem esses códigos.

Palavras Chave: Códigos lineares, códigos corretores de erros, corpos finitos.

Introdução

Claude Shannon iniciou a Teoria da Informação em 1948. Devido aos primeiros trabalhos de Shannon e aos avanços científicos nessa área na segunda metade do século XX, atualmente somos capazes de nos comunicar com facilidade e segurança pelos diversos canais de comunicações tais como: celular, internet, etc.

Quando uma mensagem é transmitida por um canal de comunicação, a mesma fica sujeita a ruídos e a outras interferências que modificam seu conteúdo, deixando-a distorcida quando chega ao seu destinatário. Observando esse problema, a solução encontrada foi adicionar redundâncias a uma mensagem num processo chamado de codificação de modo que, ao passar pelo canal de transmissão, mesmo a mensagem sofrendo um certo número de alterações, seja possível entender o seu conteúdo correto após decodificá-la (um processo inverso ao de codificação). Deste modo, quanto mais erros forem possíveis de corrigir em uma mensagem por um decodificador, melhor será, pois nessas condições haverá uma grande chance da mensagem chegar com o conteúdo correto ao destinatário. No entanto, também é importante ter eficiência computacional nesses processos de codificação e decodificação.

Para definir os códigos é preciso anteriormente dizer quais são os elementos que nos permitem escrevê-los, isto é, qual é o alfabeto que nos permite criar as 'palavras' do código. O alfabeto será sempre um corpo finito com q elementos, o qual denotamos por $GF(q)$. Da Álgebra Linear, sabemos que $GF(q)^n$ é um espaço vetorial

*Este trabalho é uma explanação à comunidade científica, resultante do projeto de iniciação científica número 2011/10345-0 financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

†E-mails, respectivamente: robcardo@ig.com.br, andrade@ibilce.unesp.br. Departamento de Matemática, Instituto de Biociências, Letras e Ciências Exatas (IBILCE/UNESP). O primeiro autor cursa Bacharelado em Matemática Pura e é orientado pelo segundo autor.

sobre o alfabeto $GF(q)$. Dessa maneira, para podermos utilizar as importantes ferramentas advindas desta álgebra, definimos código linear como sendo um subespaço vetorial de $GF(q)^n$. Portanto, na própria construção do ambiente matemático em que trabalhamos já é perceptível a importância dos corpos finitos. Sobre essas últimas estruturas algébricas é que trataremos a seguir, antes de prosseguirmos a teoria (a nível introdutório) dos códigos corretores de erros.

O presente trabalho está assim distribuído: na Seção 1 faremos um breve histórico do conceito de corpos finitos juntamente com alguns principais resultados da existência e unicidade de tais corpos; na Seção 2, apresentamos alguns resultados sobre códigos corretores de erros, assim como um diagrama de seu funcionamento; na Seção 3, apresentamos o conceito de códigos lineares enfocando seus principais parâmetros tais como matrizes geradoras, teste de paridade e código dual, e apresentamos um algoritmo de correção de erros para os códigos lineares corretores de erros.

1 Corpos finitos

Na Matemática, estamos acostumados a trabalhar com corpos infinitos ($\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$). No entanto, também existem corpos finitos, como é o caso da classe de restos módulo p (p primo), a qual denotamos por $\mathbb{Z}_p$ ou por $GF(p)$. Formalmente, um corpo finito é um conjunto com finitos elementos munido das operações de soma e multiplicação que respeitam às propriedades associativa, comutativa, existência de elemento neutro, existência do elemento inverso e distributividade da multiplicação com relação à soma. Por exemplo, $GF(2) = \{0, 1\}$ é um corpo através da soma e da multiplicação módulo 2. Esse corpo, $GF(2)$, é muito especial e é chamado de **código binário**. Em geral, denotamos por $GF(q)$ um corpo finito com q elementos.

Algumas informações muito importantes que precisamos saber sobre corpos finitos estão destacadas nos itens abaixo, que podem ser encontrados em [1].

Existência : Existe um corpo finito com q elementos se, e somente se, q é uma potência de um número primo.

Unicidade : Existe um único corpo finito para qualquer potência de um número primo, a menos de isomorfismos.

Elemento Primitivo : Se $\mathbb{K}$ é um corpo finito, então o grupo abeliano multiplicativo $\mathbb{K}^*$ é cíclico. Portanto, existe um número $\alpha \in \mathbb{K}$ tal que todo elemento de $\mathbb{K}^*$ pode ser escrito como uma potência de α. Esse elemento é chamado de elemento primitivo do corpo.

Num corpo $GF(p)$ sabemos adicionar e multiplicar seus elementos módulo p, se p é primo. Agora, seja $GF(p^m)$ um corpo finito. Devido à unicidade de corpos finitos (a menos de isomorfismos), veja que $\frac{\mathbb{Z}_p}{\langle p(x) \rangle}$ (anel quociente) é um corpo finito com p^m elementos, sendo $p(x)$ um polinômio mônico irredutível de grau m sobre $\mathbb{Z}_p$. Por isso, podemos considerar $GF(p^m)$ como sendo $\frac{\mathbb{Z}_p}{\langle p(x) \rangle}$, cujos elementos sabemos adicionar e multiplicar módulo $p(x)$. Para exemplificar a construção de um corpo finito, vamos construir o corpo $GF(16)$.

Exemplo 1 *Construção do corpo $GF(2^4) = GF(16)$. Primeiramente vamos encontrar um polinômio primitivo em $GF(2)$ (denominado polinômio primitivo um polinômio mônico irredutível em que uma de suas raízes é um elemento primitivo do corpo). Considere $p(x) = x^4 + x + 1$ um polinômio mônico irredutível sobre*

$GF(2)$. Agora, seja α uma raiz de $p(x)$, isto é, $\alpha^4 + \alpha + 1 = 0$. Vamos mostrar que todos os elementos de $GF(2^4)$ são potências de α e que são escritos como combinação linear de 1, α, α^2 *e* α^3. *De fato, como $\alpha^4 = \alpha + 1$, segue que*

α^0	$1 + 0\alpha + 0\alpha^2 + 0\alpha^3$	1000
α	$0 + 1\alpha + 0\alpha^2 + 0\alpha^3$	0100
α^2	$0 + 0\alpha + 1\alpha^2 + 0\alpha^3$	0010
α^3	$0 + 0\alpha + 0\alpha^2 + 1\alpha^3$	0001
α^4	$1 + 1\alpha + 0\alpha^2 + 0\alpha^3$	1100
α^5	$0 + 1\alpha + 1\alpha^2 + 0\alpha^3$	0110
α^6	$0 + 0\alpha + 1\alpha^2 + 1\alpha^3$	0011
α^7	$\alpha^4 + \alpha^3 = 1 + 1\alpha + 0\alpha^2 + 1\alpha^3$	1101
α^8	$\alpha^4 + \alpha^2 + \alpha = 1 + 0\alpha + 1\alpha^2 + 0\alpha^3$	1010
α^9	$0 + 1\alpha + 0\alpha^2 + 1\alpha^3$	0101
α^{10}	$\alpha^4 + \alpha^2 = 1 + 1\alpha + 1\alpha^2 + 0\alpha^3$	1110
α^{11}	$0 + 1\alpha + 1\alpha^2 + 1\alpha^3$	0111
α^{12}	$\alpha^2 + \alpha^3 + \alpha^4 = 1 + 1\alpha + 1\alpha^2 + 1\alpha^3$	1111
α^{13}	$\alpha + \alpha^2 + \alpha^3 + \alpha^4 = 1 + 0\alpha + 1\alpha^2 + 1\alpha^3$	1011
α^{14}	$\alpha + \alpha^3 + \alpha^4 = 1 + 0\alpha + 0\alpha^2 + 1\alpha^3$	1001
α^{15}	$\alpha + \alpha^4 = 1 + 0\alpha + 0\alpha^2 + 0\alpha^3$	1000

em que os vetores (a, b, c, d) são simplificadamente representados por abcd. Portanto, α é um elemento primitivo de $GF(16)$ e $p(x)$ é um polinômio primitivo. Portanto, $GF(16) = \frac{GF(2)[x]}{\langle x^4+x+1\rangle}$. Assim, a construção de $GF(16)$ está feita. Dessa maneira, já sabemos operar quaisquer elementos deste conjunto. Por exemplo, podemos encontrar o valor do produto de 0110 *por* 1110 *(observe que essa é uma representação vetorial simplificada dos vetores $(0, 1, 1, 0)$ e $(1, 1, 1, 0)$ de $GF(16)$) da seguinte maneira*

$$0110 \times 1110 = \alpha^5 \times \alpha^{10} = \alpha^{15} = 1 = 1000$$

ou ainda, podemos encontrar a soma de 1010 *e* 1111 *fazendo*

$$1010 + 1111 = 1 + 0\alpha + 1\alpha^2 + 0\alpha^3 + 1 + 1\alpha + 1\alpha^2 + 1\alpha^3 = 0 + 1\alpha + 0\alpha^2 + 1\alpha^3 = 0101.$$

2 Códigos corretores de erros

Nesta seção, apresentamos alguns resultados importantes sobre códigos corretores de erros. Um sistema de comunicação conecta uma fonte de dados a um receptor de dados através de um canal. São exemplos de canais: cabos coaxiais, circuitos telefônicos, transmissão por microondas e fitas magnéticas, que pode ser representado na figura 1.

A seguir faremos um breve histórico de como funciona um sistema de comunicação. Primeiramente, uma mensagem entra no sistema de comunicação a partir da fonte de dados escrita com letras no alfabeto $GF(q)$ em questão e é chamada de *código da fonte*. Depois, essa sequência de dígitos é convertida em outra pelo codificador, transformando-se numa sequência chamada de *código do canal*. A nova sequência é mais longa, apresentando o que chamamos de dígitos de redundância, os quais são inseridos para que seja possível identificar erros na mensagem e corrigi-los. A mensagem codificada pelo codificador no sistema de transmissão é então enviada pelo canal. Na saída, a mensagem passa pelo decodificador, que tenta identificar se ocorreram erros na transmissão da mensagem pelo canal e, neste caso, corrigí-los.

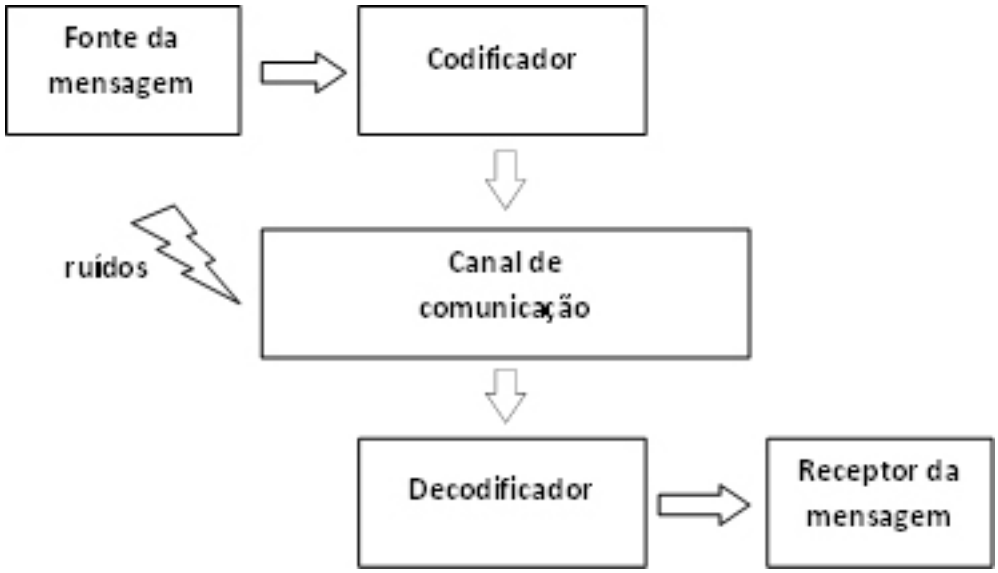


Figura 1: Representação de um sistema de comunicação

Por fim, a mensagem decodificada volta à mensagem original e é enviada ao usuário receptor da mensagem, completando o seu trajeto pelo sistema.

Para conseguir contar a quantidade de erros ocorridos em um canal precisamos identificar uma forma de medida entre vetores de um espaço vetorial $GF(q)^n$. Essa medida pode ser dada pelo número de entradas distintas desses vetores u e v, que é chamada de **distância de Hamming** e é denotada por $d(u, v)$, a qual é uma métrica. Definimos também distância mínima de um código sobre $GF(q)^n$ como sendo o valor da menor medida entre todas as palavras distintas de um código C. Sabendo isso, podemos mencionar o importante resultado a seguir:

Teorema 2 *Se C é um código com distância mínima d, então C é capaz de detectar simultaneamente até $d-1$ erros e corrigir até $[\frac{d-1}{2}]$ erros (a notação $[x]$ indica o maior número inteiro menor do que x).*

Segue como corolário desse teorema que um código que corrige até t erros deve ter distância mínima $d \geq 2t+1$.

3 Códigos lineares

Nesta seção apresentamos o conceito de códigos lineares enfocando seus principais parâmetros tais como matrizes geradoras, teste de paridade e código dual. Na segunda parte, apresentamos um algoritmo de correção de erros para os códigos lineares.

3.1 Códigos lineares

Um código linear C é um subespaço vetorial de $GF(q)^n$ sobre $GF(q)$. É importante notar que um código linear $C \subset GF(q)^n$ com dimensão k sobre $GF(q)$ tem q^k elementos. Uma maneira de representar um código linear é através de uma matriz conforme a definição abaixo.

Definição 3 (Matriz geradora) *Seja $C \subset GF(q)^n$ um código linear de dimensão k. A matriz G de dimensão $k \times n$ cujas linhas são compostas pelos vetores de uma das bases de C é chamada **matriz geradora** de C.*

Nessas condições, uma palavra $c \in GF(q)^n$ pertence a um código C se, e somente se, existe um vetor $x \in GF(q)^k$ tal que $c = xG$, onde G é matriz geradora de C.

Sendo $\langle u, v\rangle$ o produto interno usual dos vetores u e v em $GF(q)^n$, o complemento ortogonal de um código C é o conjunto $C^\perp = \{u \in GF(q)^n : \langle u, v\rangle = 0, \forall v \in C\}$. Todo elemento de $GF(q)^n$ é soma de um elemento de C e de um elemento de $C^\perp$. Além disso, $C^\perp$ é um subespaço vetorial de $GF(q)^n$ e, exceto o *zero*, nenhum elemento deste conjunto está em C e vice-versa. Portanto, $C^\perp$ é um código linear chamado de **código dual** de C. Além do mais, a matriz H geradora deste código é denominada **matriz teste de paridade** de C. Um resultado importante é que $c \in C$ se, e somente se, $Hc^t = 0$.

Exemplo 4 *Exemplo de um código linear em $GF(2)^4$. Sobre o alfabeto $GF(2)$, queremos transmitir as mensagens NORTE (00), SUL (01), LESTE (10) e OESTE (11). Para isso, a mensagem $u = u_1u_2$ vamos adicionar dois dígitos de redundância, criando palavras $x = x_1x_2x_3x_4$ em que $x_1 = u_1$, $x_2 = u_2$, $x_3 = u_1$ e $x_4 = u_1 + u_2$. Deste modo, obtemos*

$$C = \{0000, 0101, 1011, 1110\}$$

que é um subespaço vetorial de $GF(2)^4$ de dimensão 2. Portanto, C é um código linear. Sua matriz geradora é dada por

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

Sua matriz teste de paridade é dada por

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

3.2 Decodificação de códigos lineares

A seguir, vamos descrever um algoritmo que corrige erros na transmissão de mensagens de um código C. Isto é, recebido um vetor $v \in GF(q)^n$, o decodificador tentará, através do algoritmo, detectar os erros ocorridos no canal de transmissão, corrigí-los quando for possível e enviar ao destinatário a palavra correta. O tipo de decodificador que trataremos é incompleto. Neste caso, se o número de erros ocorridos for maior do que o esperado em um código, o decodificador não fará a decodificação, no intuito de evitar equívocos.

Sendo $C \subset GF(q)^n$ um código linear de dimensão k, para todo $v \in GF(q)^n$, o conjunto $v + C = \{v + c : c \in C\}$ é chamado classe lateral de C. Todo vetor de $GF(q)^n$ está em uma, e só em uma, dessas classes. Além disso, tem-se também que cada classe possui q^k elementos. Chama-se vetor líder de uma classe o vetor que tem mais entradas nulas dentre todos os vetores desse conjunto.

Exemplo 5 *Considere o código do Exemplo 4 sobre $GF(2)^4$ dado por*

$$C = \{0000, 0101, 1011, 1110\}.$$

Suas classes laterais são dadas por

$$C_1 = \{1000, 0011, 1101, 0110\}$$

$$C_2 = \{0001, 1010, 0100, 1111\}$$
$$C_3 = \{0010, 1001, 0111, 1100\}$$

e o vetor líder de classe classe é o primeiro elemento inserido à esquerda nesses conjuntos.

Chama-se **síndrome** de um vetor $v \in GF(q)^n$ o vetor $s = vH^t$, onde H é a matriz teste de paridade do código. Um fato importante é que dois vetores estão na mesma classe se, e somente se, têm mesma síndrome. De fato, dados dois vetores $u, v \in GF(q)^n$, tem-se que

$$u + C = v + C \iff u - v \in C \iff (u - v)H^t = 0 \iff uH^t = vH^t.$$

Portanto, podemos fazer uma tabela associando o vetor líder de cada classe com sua síndrome.

Exemplo 6 *Através do Exemplo 5 tem-se que*

Líder	0000	1000	0001	0010
Síndrome	00	11	01	10

Ao ser enviada uma palavra $c \in C$ por um canal de transmissão, os erros ocorridos podem ser descritos pelo vetor e, que faz a palavra se modificar num novo vetor $y = c + e \in GF(q)^n$. Algo importante a se notar é que a síndrome da palavra y recebida pelo decodificador é a mesma do vetor erro e. De fato, lembrando que

$$c \in C \iff cH^t = 0$$

segue que

$$eH^t = (y - c)H^t = yH^t - cH^t = yH^t.$$

Essas observações ajudam a justificar o funcionamento do algoritmo de decodificação de códigos lineares que será descrito a seguir. Abaixo, considere d a distância mínima do código $C \subset GF(q)^n$.

Algoritmo de Decodificação de códigos lineares
Entrada: $y \in GF(q)^n$ vinda do canal de comunicação.
Saída: Uma palavra c em C ou a mensagem "Não foi possível decodificar, por excesso de ruídos".

Passos:

1. Encontre a síndrome s de y.
2. Se $s = 0$, faça $c = y$ e pare. Caso contrário, prossiga.
3. Dentre as classes laterais, tome o vetor líder e cuja síndrome é s.
4. Se o número de entradas não nulas de e é menor ou igual a $[\frac{d-1}{2}]$, faça $c = y - e$ e pare. Caso contrário, escreva a mensagem "Não foi possível decodificar, por excesso de ruídos".

Exemplo 7 *No Exemplo 5 suponha que o destinatário receba a seguinte mensagem* $y = 0100$ *para ser decodificada. Aplicando o algoritmo tem-se que:*

1. *A síndrome de* y *é* $s = 01$.
2. $s \neq 0$. *Portanto, sigamos.*
3. *Da tabela criada anteriormente, o vetor líder de síndrome* 01 *era* 0001.
4. *Agora, veja que o número de entradas não nulas de* e *é* $1 > 0 = [\frac{d-1}{2}]$, *pois a distância mínima do código é* $d = 1$. *Logo, o decodificador responderá "Não foi possível decodificar, por excesso de ruídos".*

4 Conclusão

Vimos neste trabalho que adicionando certas redundâncias a uma mensagem que se deseja transmitir antes que ela passe pelo canal de comunicação, mesmo ela sofrendo no máximo um número previsto de distorções, ainda será possível recuperá-la. No entanto, precisa estar claro que não é de qualquer maneira que se adicionam essas redundâncias. É preciso de uma regra bem estabelecida de codificação que possua um processo inverso computacionalmente viável (decodificação). Nesse sentido, neste trabalho tratamos dos códigos lineares, que são um tipo importante de códigos corretores de erros e que facilitam esses processos digitais. Existem outros códigos corretores de erros e estudos com o intuito de minimizar esses problemas na transmissão de mensagens, uma vez que eliminar a ocorrência de ruídos em um canal de transmissão é um problema geralmente muito mais difícil (ou até, impossível). Dentro dos códigos lineares, existem classes de códigos corretores de erros muito utilizadas na prática, como os códigos cíclicos, códigos BCH, códigos Reed-Solomon, entre outros.

Referências

[1] Blahut, R.E. *Theory and Practice of Error Control Codes*. Addison-Wesley Publishing Company, London (1984).

[2] Hefez, A., Villela, M. L. T. *Códigos corretores de erros*, IMPA, Rio de Janeiro, (2002), São Paulo (2003).

[3] MacWilliams, F.J., Sloane, N.J.A. *The Theory of Error-Correcting Codes*. North-Holland, New York (1988).

[4] Pless, V. *Introduction to the Theory of Error-Correcting Codes*. John Wiley and Sons, New York (1989).