Correção de erros: pesos e distâncias

Na postagem Correção de erros, fiz uma pequena introdução a esse fascinante e importante tópico. Agora vou apresentar alguns conceitos para auxiliar a compreensão de algumas propriedades gerais dos códigos lineares binários. Já apresentei a notação binária e a aritmética módulo 2 que deve ser utilizada. Também falei sobre como somar palavras de código e como combiná-las linearmente, além de ter definido o produto escalar entre duas delas. Chegou a hora de definir o tamanho de um código C e o peso de um vetor pertencente a um espaço vetorial binário. Com esse conceito de peso, vou apresentar também a definição da distância de Hamming entre dois desses vetores.

Quando temos k vetores linearmente independentes formando a base de um código binário C, podemos combiná-los linearmente para produzir todas as possíveis palavras do código C. Os únicos escalares na aritmética módulo 2 que podemos usar é 0 e 1 e, portanto, uma combinação linear desses k vetores consiste de uma soma de um subconjunto da base. Assim, sejam $u_1, u_2, \ldots, u_k$ os vetores da base. Uma palavra w qualquer do código é, portanto, a combinação linear

$$w = \alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_k u_k, \qquad (1)$$

onde $\alpha_1, \alpha_2, \ldots, \alpha_k$ são escalares, cada qual podendo ser apenas 0 ou 1. Assim, se $k = 1$, temos apenas duas palavras possíveis, isto é, $w_1 = 0$ e $w_2 = u_1$. Se $k = 2$, temos quatro possíveis palavras: $w_1 = 0$, $w_2 = u_1$, $w_3 = u_2$ e $w_4 = u_1 + u_2$. Veja que ao aumentar k de uma unidade, multiplicamos por 2 o número anterior de palavras do código. Então, quando $k = 3$, devemos multiplicar 4 por 2 e obter oito palavras. De fato, quando $k = 3$, as possíveis palavras são:

$$w_1 = 0,$$

$$w_2 = u_1,$$

$$w_3 = u_2,$$

$$w_4 = u_1 + u_2,$$

$$w_5 = u_3,$$

$$w_6 = u_1 + u_3,$$

$$w_7 = u_2 + u_3$$

e

$$w_8 = u_1 + u_2 + u_3,$$

ou seja, oito palavras. Logo, para k vetores linearmente independentes, o código tem 2^k palavras no total. Portanto, dizemos que o tamanho de um código C, formado por k vetores linearmente independentes, é dado por 2^k. Note que no contexto deste parágrafo, quando escrevo $w = 0$, refiro-me ao vetor nulo, que, para um espaço vetorial de n dimensões, corresponde ao vetor $00\ldots0$, com n bits nulos.

Vamos ver um exemplo. Considere o caso de um código $[5, 3]$. Vamos tomar três vetores de cinco dimensões, linearmente independentes, tais como: $u_1 = 01101$, $u_2 = 01010$ e $u_3 = 10100$. De acordo com nossa análise acima, devemos ter 2^3, ou oito, palavras distintas. De fato,

$$w_1 = 0u_1 + 0u_2 + 0u_3 = 00000,$$

$$w_2 = 1u_1 + 0u_2 + 0u_3 = 01101,$$

$$w_3 = 0u_1 + 1u_2 + 0u_3 = 01010,$$

$$w_4 = 0u_1 + 0u_2 + 1u_3 = 10100,$$

$$w_5 = 1u_1 + 1u_2 + 0u_3 = 01101 + 01010 = 00111,$$

$$w_6 = 1u_1 + 0u_2 + 1u_3 = 01101 + 10100 = 11001,$$

$$w_7 = 0u_1 + 1u_2 + 1u_3 = 01010 + 10100 = 11110$$

e

$$w_8 = 1u_1 + 1u_2 + 1u_3 = 01101 + 01010 + 10100 = 00111 + 10100 = 10011.$$

O peso de um vetor que pertence a um espaço vetorial binário é dado pelo número de uns que tem. Assim, por exemplo, no caso de um espaço vetorial binário de cinco dimensões, o peso da palavra 01101 é 3, o peso de 01010 é 2 e o peso de 10100 também é 2. Como mais um exemplo, no caso de um espaço de 15 dimensões, o peso de 111110000011010 é 8.

Dados dois vetores de um espaço vetorial binário de n dimensões, a distância de Hamming entre eles é definida como sendo o peso de sua soma. Assim, por exemplo, dados $u_1 = 01101$, $u_2 = 01010$ e $u_3 = 10100$, a distância entre u_1 e u_2 é o peso de $u_1 + u_2 = 01101 + 01010 = 00111$, que é igual a 3. Nesse caso, escrevemos $d(u_1, u_2) = 3$. Veja que o cálculo do peso e da distância de Hamming é feito com aritmética usual, mas **não** é feito através da aritmética módulo 2. Observe que $d(u, v) = d(v, u)$ e que $d(u, v) \geqslant 0$, sendo que se $d(u, v) = 0$, então $u = v$, para quaisquer u e v pertencentes ao espaço vetorial. Além disso, a desigualdade triangular também pode ser facilmente verificada: $d(u, v) \leqslant d(u, w) + d(w, v)$, para quaisquer u, v e w pertencentes ao espaço vetorial.

Um conceito muito importante para podermos demonstrar os teoremas das próximas postagens sobre este tópico de correção de erros é o da distância mínima entre duas palavras distintas de um código, sendo que a distância mínima nula de um código não é definida. Veja que o conjunto de todas as somas entre duas palavras do código é igual ao conjunto de todas as palavras do código. Sendo assim, listando todas as somas entre duas palavras quaisquer do código, estaremos listando todas as palavras do código. Logo, a menor distância entre duas palavras distintas do código é idêntica ao menor peso entre os pesos das palavras não nulas do código.

Como exemplo, consideremos de novo o código $[5, 3]$ definido pelos três vetores de base: $u_1 = 01101$, $u_2 = 01010$ e $u_3 = 10100$. Como vimos, as oito palavras desse código são 00000, 01101, 01010, 10100, 00111, 11001, 11110 e

10011. Os pesos dessas palavras são, respectivamente, 0, 3, 2, 2, 3, 3, 4 e 3. Logo, o menor peso não nulo é 2, que é também a menor distância entre duas palavras distintas do código.