Sílvio A. Abrantes

Livro de receitas. Receitas?!

Uns pequenos truques que facilitam alguns cálculos de Códigos e Teoria da Informação

FEUP

Codificação aritmética:
Representação binária de números reais positivos inferiores a 1

Problema a resolver: dado um número real r entre 0 e 1, qual é a sua representação binária se r for dado como uma fracção ou como um número decimal?

Há várias maneiras de resolver o problema, que passa por exprimir o número *r* na forma $r = \sum_{j=1}^{m} b_j 2^{-j}$, em que os coeficientes b_j são binários (0 ou 1). A partir daí a solução é imediata:

$$r = \sum_{j=1}^{m} b_j 2^{-j} = (.b_1 b_2 \ldots b_m)_2 \quad \Rightarrow \quad b_1 b_2 \ldots b_m$$

em que $b_1 b_2 \ldots b_m$ é a representação binária pretendida.

Se o número *r* for uma fracção em que o denominador é uma potência de 2 (fracção diádica) a representação binária é muito simples: acha-se a representação binária do numerador com um número de bits igual ao expoente da potência de 2. Por exemplo, a expansão binária de 13/64 = $13/2^6$ é $(.001101)_2$, ou 001101, porque $13=1101_2$ e precisamos de 6 bits.

E se o denominador não for uma potência de 2? Nesse caso pode não ser muito fácil ou imediato exprimir o número *r* através daquele somatório.

Receita[1]:

Duplicar o número. Sempre que o resultado for menor que 1 o dígito binário correspondente é 0, senão é 1. Neste último caso subtrai-se 1 ao resultado e prossegue-se com as duplicações.

[1] De acordo com Neal Koblitz, *A Course in Number Theory and Cryptography*, Springer Verlag, 2ª edição, 1994.

Exemplos

1) Fracção: $r = \frac{2}{7}$

Vamos construir uma tabela com os passos a dar:

Duplicações e ajustes	$4/7$	$8/7$	$2/7$	$4/7$	$8/7$	$2/7$	...
Bits	0	1	0	0	1	0	...

Na quarta coluna a fracção $2/7$ foi obtida assim: $(8/7-1)\times 2$.

O padrão 010 repete-se indefinidamente (representemos esse facto por $\overline{010}$). A representação binária de $2/7$ é, portanto,

$$\frac{2}{7} = \left(.\overline{010}\right)_2 \quad \Rightarrow \quad 010010010\ldots$$

2) Decimal: $r = 0{,}8776$

Poderíamos construir uma tabela análoga à anterior, disposta na horizontal. Em vez disso façamo-lo na vertical para melhor visualizar as duplicações:

Duplicações e ajustes	Bits
1,7552	1
1,5104	1
1,0208	1
0,0416	0
0,0832	0
0,1664	0
0,3338	0
0,6676	0
1,3352	1
0,6704	0
1,3408	1

O processo prosseguiria como até aqui. A representação binária de 0,8776 é, portanto, 11100000101….

Codificação aritmética:
Como representar um intervalo com o menor número de bits?

Ou: "Como calcular fracções diádicas com o menor denominador"[2]

Queremos representar um intervalo [α, β[de largura *L* por uma fracção diádica *r* para daí se obter a menor representação binária (representação binária com menos bits).

Uma fracção diádica é um número racional da forma

$$r = \frac{q}{2^m} \qquad (q \text{ – inteiro; } m \text{ – inteiro não negativo})$$

Trata-se de um caso especial das fracções p-ádicas q/p^m, em que *p* é um número primo.

Estamos interessados em obter uma fracção diádica que tenha o menor denominador possível. Para isso *q* deverá ser ímpar e *m* o menor possível. Sendo os extremos do intervalo não-negativos, *q* será também não-negativo.

Mas porquê uma fracção diádica? Porque a sua representação ou expansão binária é muito fácil de calcular: repare-se que sendo *q* inteiro e inferior a 2^m podemos escrever

$$r = \frac{q}{2^m} = \frac{1}{2^m}\sum_{j=1}^{m} b_j 2^{m-j} = \sum_{j=1}^{m} b_j 2^{-j} = (.b_1 b_2 \ldots b_m)_2 \quad \Rightarrow \quad b_1 b_2 \ldots b_m$$

onde os *m* coeficientes b_j são binários (0 ou 1) e $b_1 b_2 \ldots b_m$ representa a expansão binária de *r*. Vê-se então que o expoente *m* é o número de bits a reter na representação binária do numerador *q*. Assim, a expansão binária de 11/32, por exemplo, é imediata: com $2^5 = 32$ precisamos de 5 bits para representar $11 = 1011_2$, isto é, $(.01011)_2$ ou 01011.

Seja então o intervalo [α, β[. Qual o melhor representante binário deste intervalo (no sentido de necessitar de menos bits)? Por exemplo, qual é a palavra binária que representa com menos bits o intervalo [0,352; 0,493[?

Há dois métodos de cálculo da fracção diádica adequada:

---

[2] De acordo com Hankerson, Harris e Johnson, *Introduction to Information Theory and Data Compression*, CRC Press, 1998, pág. 123.

Método 1:

Passo 1: Encontrar o *menor* inteiro *t* tal que $\frac{1}{2^t} \le L = \beta - \alpha$ ou $2^t \ge 1/L$.

(é equivalente a determinar o inteiro t tal que $2^{-t} \le L < 2^{-t+1}$ *ou que* $t = \lceil -\log_2 L \rceil$*)*

Passo 2: Resolver a dupla inequação $\alpha \le \frac{x}{2^t} < \beta$ para inteiros x. Há um inteiro, no máximo dois, que satisfaz a dupla inequação. Se houver dois inteiros, serão consecutivos e deverá escolher-se o inteiro par. Em qualquer caso $r = \frac{x}{2^t}$, simplificando com os menores termos possíveis, é a fracção diádica com menor denominador no intervalo [α, β[.

Método 2:

Expandem-se α e β na forma binária até que as palavras binárias comecem a ser diferentes (em α o primeiro bit diferente é um "0" e em β é um "1"). Tomam-se os bits comuns e acrescenta-se um "1".

O problema com este método é que há duas excepções: quando pelo menos um dos extremos, α ou β, já é ele próprio uma fracção diádica.

Exemplos

Exemplo 1 com o método 1: *Intervalo* [0,4936; 0,5008[, *largura* 0,0072

Vai ser 1/2 pois 0,5 = 1/2 pertence ao intervalo e 1/2 é a sua fracção diádica com menor denominador. Mas se não soubéssemos? Aplicando o método 1:

Passo 1: encontrar o menor inteiro *t* tal que $\frac{1}{2^t} \le L = \beta - \alpha = 0,0072$ (ou que $2^t \ge 1/L = 138,9$), ou o inteiro *t* que satisfaça $2^{-t} \le 0,0072 < 2^{-t+1}$. A solução é *t = 8*.

Passo 2: agora determinamos o inteiro *x* em $\alpha \le \frac{x}{2^t} < \beta$, isto é, $0,4936 \le \frac{x}{2^8} < 0,5008$. A solução, dupla, é $x = 127$ e $x = 128$. Escolhe-se $x = 128$ por ser par (é que assim vai permitir simplificar a fracção).

A fracção pretendida é, finalmente, $r = \frac{x}{2^t} = \frac{128}{256} = \frac{1}{2}$, como se esperava. A representação binária é $(.1)_2$ ou, desprezando o ponto, simplesmente 1.

Exemplo 2 com o método 1: *Intervalo* [0,876; 0,8776[, *largura* 0,0016

Passo 1: encontrar o menor *t* que satisfaz $\frac{1}{2^t} \leq 0,0016$ ou, de modo equivalente, *t* que satisfaz $2^{-t} \leq 0,0016 < 2^{-t+1}$. A solução é *t* = 10.

Passo 2: resolver $0,876 \leq \frac{x}{2^{10}} < 0,8776$, isto é, *x* = 898 (só há uma solução inteira). Assim, a fracção diádica com menor denominador é:

$$r = \frac{x}{2^t} = \frac{898}{1024} = \frac{449}{512}$$

Daqui se tira a representação binária que desejamos:

$$\frac{449}{512} = \frac{256+128+64+1}{512} = \frac{1\times 2^8 + 1\times 2^7 + 1\times 2^6 + 1\times 2^0}{512} = (.111000001)_2$$

Representação binária: 111000001.

Exemplo 2 com o método 2: *Intervalo* [0,876; 0,8776[, *largura* 0,0016

- Representação binária de α:

$$\alpha = 0,876 = (.11100000010\ldots)_2$$

- Representação binária de β:

$$\beta = 0,8776 = (.11100000101\ldots)_2$$

- Nem α nem β são fracções diádicas (não são dízimas finitas) pelo que o intervalo dado não faz parte das excepções do método 2.

Os bits comuns a α e β são 11100000. Acrescentando um bit "1" obtemos $r = (.111000001)_2$, como há pouco. Logo, a melhor representação binária do intervalo é: 111000001.

Situações de excepção do método 2

- Se $\alpha = (.a_1 a_2 \ldots a_{t-1})_2$ a expansão é finita, isto é, α é uma fracção diádica. Nesse caso $r = \alpha$.

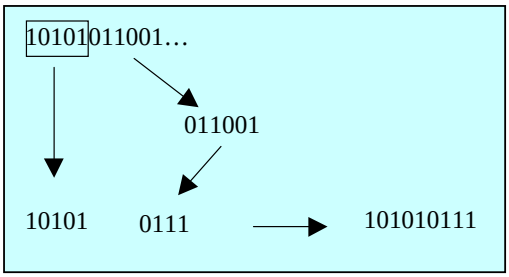
- Se $\alpha > (.a_1 a_2 \ldots a_{t-1})_2$ e $\beta = (.a_1 a_2 \ldots a_{t-1}1)_2$

(isto é, $\alpha = (.a_1 a_2 \ldots a_{t-1} 0 \ldots 1 \ldots)_2$ *e* $\beta = (.a_1 a_2 \ldots a_{t-1} 1000 \ldots)_2$ *é uma fracção diádica)*

Em α o bit 0 a seguir ao bit a_{t-1} é o primeiro bit diferente de β. Depois desse bit 0 há-de aparecer um bit 1. Se depois deste apenas aparecerem zeros toma-se $r = \alpha$; se não, o primeiro zero depois do 1 passa a 1 e trunca-se a expansão aí, originando *r*.

Por exemplo, se $\alpha = (.101010111)_2$ e $\beta = (.101011)_2$ (primeira excepção), toma-se $r = (.101010111)_2$ imediatamente.

Se $\alpha = (.10101011001\ldots)_2$ e $\beta = (.101011)_2$ (segunda excepção) os bits comuns aos dois números são 10101 e em α aparecem mais alguns "uns" (10101**11**...). Se depois destes "uns" os bits fossem todos 0 a expansão seria $r = (.10101011)_2$ mas como não é assim, o bit 0 que surge logo a seguir passa a 1 e a expansão binária procurada fica encontrada: $r = (.101010111)_2$, como se ilustra na figura seguinte.



Códigos correctores de erros:
"Receita" para multiplicar um vector binário por uma matriz binária

Receita:

Somam-se as linhas de ordem i da matriz, em que i é a posição dos "uns" no vector.

Exemplos de aplicação:

1) codificação de códigos de blocos binários

A palavra de código **Y** é obtida multiplicando o vector de informação **X** pela matriz geradora **G**: **Y = XG**.

2) descodificação de códigos de blocos binários: cálculo da síndrome

O vector síndrome é obtido multiplicando a palavra codificada **Z** pela matriz de verificação de paridade **H**: **S = ZH**.

Exemplo numérico de geração de palavras de código:

Seja $\mathbf{X} = [x_1 x_2 x_3 x_4] = [1011]$ e a matriz geradora de um código (7, 4)

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Com esta matriz geradora a palavra de código **Y** correspondente a **X** é igual a **Y** = **XG** = [1011$c_1c_2c_3$]. Quanto vale **Y**?

Uma maneira de calcular os três bits de paridade **C** = [$c_1c_2c_3$] é fazer a multiplicação matricial habitual linha-coluna, obtendo-se as chamadas *equações de paridade*,

$$\begin{cases} c_1 = x_1 + x_2 + x_4 \\ c_2 = x_1 + x_3 + x_4 \\ c_3 = x_1 + x_2 + x_3 \end{cases}$$

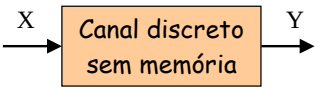
de que resultaria $\mathbf{C} = [1+1 \quad 1+1+1 \quad 1+1] = [0 \quad 1 \quad 0]$, ou seja, $\mathbf{Y} = [1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0]$.

Uma maneira mais rápida é somar simplesmente as linhas 1, 3 e 4 de **G:**

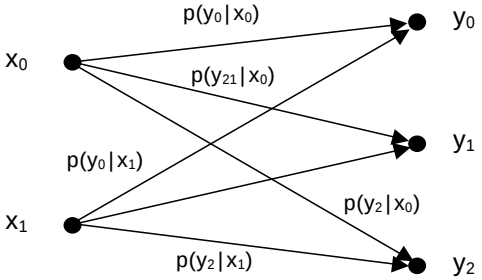
	1	0	0	0	1	1	1	(1)
	0	0	1	0	0	1	1	(3)
$\oplus$	0	0	0	1	1	1	0	(4)
Y =	[1	0	1	1	0	1	0]	

Teoria da Informação:
Simplificação de canais discretos compostos

Um canal discreto sem memória como a da figura seguinte pode ser representado pelo diagrama de probabilidades de transição ou pela matriz de transição.

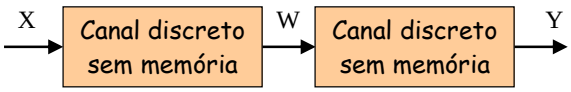


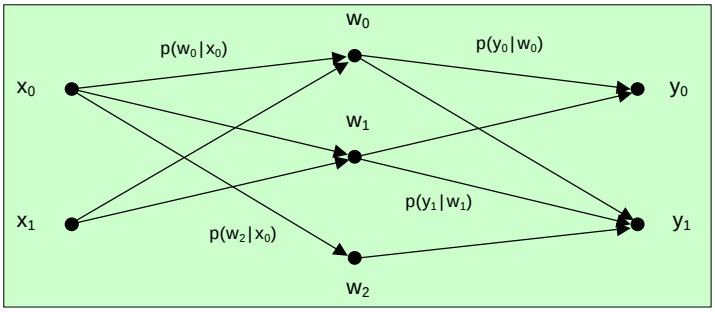
Assim, por exemplo, se à entrada tivermos um sinal discreto X com dois valores possíveis, x_0 e x_1, e à saída tivermos Y com três valores possíveis, y_0, y_1 e y_2, o diagrama de probabilidades de transição e a matriz de transição são como se apresentam a seguir.



$$[P(Y \mid X)] = \begin{bmatrix} P(y_0 \mid x_0) & P(y_1 \mid x_0) & P(y_2 \mid x_0) \\ P(y_0 \mid x_1) & P(y_1 \mid x_1) & P(y_2 \mid x_1) \end{bmatrix}$$

E se tivermos dois canais em série como nas figuras seguintes? Como determinar o canal global equivalente $X \rightarrow Y$?





Há duas maneiras:

1. A partir das matrizes de transição:

Receita:

A matriz de transição global é igual ao produto das matrizes de transição individuais.

2. A partir do diagrama de probabilidades de transição:

Receita:

As probabilidades condicionais $p(y_j|x_i)$ *(*x_i *– entrada;* y_j *– saída) são iguais à soma das probabilidades associadas aos diversos "trajectos" de* x_i *para* y_j*.*

Exemplo 1 (com matriz):

Na figura anterior seja

$$[P(W|X)] = \begin{bmatrix} P(w_0|x_0) & P(w_1|x_0) & P(w_2|x_0) \\ P(w_0|x_1) & P(w_1|x_1) & P(w_2|x_1) \end{bmatrix} = \begin{bmatrix} 0,2 & 0,3 & 0,5 \\ 0,4 & 0,5 & 0,1 \end{bmatrix}$$

e

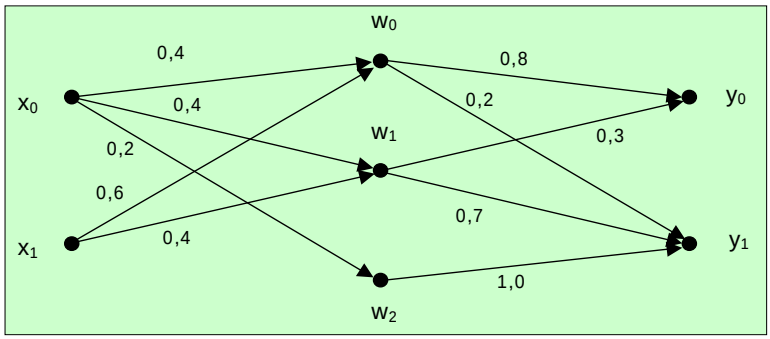
$$[P(Y|W)] = \begin{bmatrix} 0,6 & 0,4 \\ 0,5 & 0,5 \\ 0,7 & 0,3 \end{bmatrix}$$

A matriz global do canal binário resultante é dada pelo produto das duas matrizes:

$$[P(Y|X)] = [P(W|X)] \times [P(Y|W)] = \begin{bmatrix} 0,62 & 0,38 \\ 0,56 & 0,44 \end{bmatrix}$$

Exemplo 2 (com diagrama):

Seja o canal composto seguinte:



- Há dois percursos de x_0 para y_0:

$$\begin{matrix} x_0 \to w_0 \to y_0 \\ x_0 \to w_1 \to y_0 \end{matrix} \Rightarrow \quad P(y_0 \mid x_0) = \underbrace{P(w_0 \mid x_0)P(y_0 \mid w_0)}_{x_0 \to w_0 \to y_0} + \underbrace{P(w_1 \mid x_0)P(y_0 \mid w_1)}_{x_0 \to w_1 \to y_0} = 0,4 \times 0,8 + 0,4 \times 0,3 = 0,44$$

- Há dois percursos de x_1 para y_0:

$$\begin{matrix} x_1 \to w_0 \to y_0 \\ x_1 \to w_1 \to y_0 \end{matrix} \Rightarrow \quad P(y_0 \mid x_1) = P(w_0 \mid x_1)P(y_0 \mid w_0) + P(w_1 \mid x_1)P(y_0 \mid w_1) = 0,6 \times 0,8 + 0,4 \times 0,3 = 0,6$$

- Há três percursos de x_0 para y_1:

$$\begin{matrix} x_0 \to w_0 \to y_1 \\ x_0 \to w_1 \to y_1 \\ x_0 \to w_2 \to y_1 \end{matrix}$$

Logo, a probabilidade $P(y_1 \mid x_0)$ é a soma das probabilidades associadas aos três percursos.

No caso que falta, de x_1 para y_1, far-se-ia de modo semelhante. O resultado final seria o canal binário seguinte: