1 CÓDIGOS CORRECTORES

Piensa en un número entre 0 y 15. Si siempre dices la verdad, yo podría adivinar tu número con 4 preguntas, cuyas posibles respuestas son: "sí" o "no". ¿Por qué?

Un truco para justificar lo anterior es representar cada número n entre 0 y 15 en sistema binario: $n = a \cdot 2^3 + b \cdot 2^2 + c \cdot 2 + d$, donde a, b, c, d son o 0 o 1. Es decir, comunicar n es lo mismo que comunicar los cuatro bits a, b, c, d. Por tanto, se requieren de cuatro preguntas para determinar el número.

Ejercicio 1: ¿Cuáles podrían ser las preguntas?

Pero si ahora complicamos un poco más el problema y se permite mentir una (sola) vez. ¿Cuántas preguntas te tendría que hacer para detectar si mientes? ¿Cuántas en el caso de que quiera *corregir* tu mentira (o error) y adivinar el número que realmente pensaste?

Canal binario perfecto

Un canal binario es un canal para comunicar bits. En el primer problema, tenemos un canal binario perfecto para comunicar bloques de 4 bits entre tú y yo. "Perfecto" porque el canal transmite el mensaje sin cambiarlo.

Canales con ruido

En la práctica, los canales binarios no son perfectos, y pueden cambiar un 1 en un 0, o viceversa, con cierta probabilidad (Teoría de Shannon). El segundo problema, cuando se permite una mentira, es un ejemplo de esta situación: la mentira intercambia los bits. La teoría de los codigos correctores de error, o teoría de la codificación, trata de métodos de procesar mensajes para proteger contra errores introducidos por el canal.

La idea fundamental es introducir cierta estructura o información redundante en el mensaje que se envía. Si esa estructura no está en el mensaje que se recibe, se sabe que ha habido cambios en el canal; una estructura lo suficientemente rica permite recuperar el mensaje original, siempre que no haya habido demasiados cambios. Es muy parecido a lo que ocurre en lenguaje natural: piensa en cómo funcionan los programas correctores de ortografía. Imaginemos que recibimos el siguiente mensaje de texto:

"En un lular de la Mancha"

Nos damos cuenta inmediatamente de que se han producido errores en la transmisión, porque "lular" no es una palabra del castellano. Ésta es una idea importante: *no todas las combinaciones de letras son palabras válidas de nuestro diccionario* y esto nos permite detectar en este caso el error.

Pero vayamos más allá: supongamos que transmitimos la palabra "Zaragoza" y recibimos, por ejemplo, "Zatagoza". Por supuesto, detectamos que se ha producido algún error; pero aún más, cualquiera se sentiría en disposición de corregir el error: se ha producido en el tercer símbolo, y era una r en lugar de una t. La razón es clara: no hay palabras en castellano "cerca" (en el sentido de "parecidas") de Zaragoza. Pero si transmitimos "casa" y recibimos "cusa",

pese a que detectamos el error, ya no está tan claro cómo corregirlo: podríamos haber emitido lusa, musa, cuna, etc. Peor aún, podríamos haber recibido "tasa" en lugar de "casa" y ni siquiera podríamos detectar el error. La razón, la misma de antes, pero al revés: ahora hay muchas palabras semejantes ("muy cerca") a "casa".

Enumeremos las enseñanzas de estos ejemplos:

- La estructura: un conjunto de símbolos (el abecedario) y unas palabras formadas con ellos (el diccionario).
- Las palabras del diccionario deben estar separadas (para detectar errores) . . . y si están muy separadas, hasta nos atreveremos a corregir.

Volviendo a nuestro problema de adivinar un número, un caso más fácil es reconocer un mentiroso, sin intentar recuperar su número. Para esto, en vez de mandar 4 bits $[a, b, c, d]$, se mandan 5, $[a, b, c, d, e]$ donde e se escoge de manera que el número total de 1's en el vector sea par. Si los bits del mensaje que llega no satisfacen la condición de paridad, se detecta que ha habido errores, pero no hay manera de corregirlos.

Ejercicio 2: Continuando con el ejercicio 1, ¿cuál sería la pregunta para determinar e?

Para corregir, y no solamente detectar, errores, hay que introducir aún más estructura en el mensaje.

1.1 Códigos de Hamming

Un *código lineal* de longitud n y rango k es un subespacio lineal C con dimensión k del espacio vectorial F_2^n, donde $F_2 = \{0, 1\}$ (es el cuerpo finito con 2 elementos). Tal código se denomina código binario. Los vectores en C se llaman *palabras* de código. El *tamaño* de un código es el número de palabras del código y es igual a $2k$.

El *peso* w de una palabra del código es el número de sus elementos que son distintos de cero y la *distancia entre dos palabras* del código es la *distancia de Hamming* entre ellos, es decir, el número de elementos en los que difieren. La *distancia* d de un código lineal es el peso mínimo de sus palabras del código distintas de cero, o de forma equivalente, la distancia mínima entre palabras del código diferentes. Un código lineal de longitud n, dimensión k, y distancia d se denomina $[n, k, d]$ código.

Más definiciones: Una matriz G de orden $k \times n$ cuyas filas forman una base de C se llama *matriz generadora*. Una matriz H de orden $(n - k) \times n$ tal que el producto de H por cualquier elemento de C resulte el vector nulo de llama *matriz de comprobación de paridad*. Todo elemento del código C es de la forma xG, donde x es un vector en F_2^n.

Las demostraciones de las siguientes propiedades de los códigos lineales se pueden encontrar en cualquier libro introductorio de Teoría de Códigos:

1. Existe una matriz generadora G de C de la forma (I_k, A) donde I_k es la matriz identidad de orden k y A es una matriz de orden $k \times (n-k)$.

2. Existe una matriz de comprobación de paridad H para C de la forma (B, I_{n-k}), donde B es una matriz de orden $(n-k) \times k$. Más aún, $B = A^t$.

3. Un código de distancia mínima d detecta $d-1$ errores y corrige $[(d-1)/2]$ errores.

En nuestro problema de adivinar un número, consideraremos el código de Hamming $[7, 4, 3]$ (el hecho que $d = 3$ es una consecuencia de la definición del código). Según la propiedad 3, detecta dos errores y corrige uno. Una matriz generadora sería

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

Observa que si $x = \begin{pmatrix} a & b & c & d \end{pmatrix}$, entonces

$$xG = \begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \\ d \\ a+b+d \\ a+c+d \\ b+c+d \end{pmatrix},$$

que es una relación que nos será útil para construir explícitamente el código. Veamos cómo la matriz G genera las palabras del código que incluye los (tres)

dígitos redundantes para cada número del 0 al 15:

	$\mathbf{2}^0$	$\mathbf{2}^1$	$\mathbf{2}^3$	$\mathbf{2}^4$	$\mathbf{2}^5$	$\mathbf{2}^6$	$\mathbf{2}^7$		P_1	P_2	P_3	P_4	P_5	P_6	P_7
0	0	0	0	0	0	0	0								
1	1	0	0	0	1	1	0		1				1	1	
2	0	1	0	0	1	0	1			2			2		2
3	1	1	0	0	0	1	1		3	3				3	3
4	0	0	1	0	0	1	1				4			4	4
5	1	0	1	0	1	0	1		5		5		5		5
6	0	1	1	0	1	1	0			6	6		6	6	
7	1	1	1	0	0	0	0		7	7	7				
8	0	0	0	1	1	1	1	$\longrightarrow$				8		8	8
9	1	0	0	1	0	0	1		9			9			9
10	0	1	0	1	0	1	0			10		10		10	
11	1	1	0	1	1	0	0		11	11		11	11		
12	0	0	1	1	1	0	0				12	12	12		
13	1	0	1	1	0	1	0		13		13	13		13	
14	0	1	1	1	0	0	1			14	14	14			14
15	1	1	1	1	1	1	1		15	15	15	15	15	15	15
α	a	b	c	d	e	f	g		y_1	y_2	y_3	y_4	y_5	y_6	y_7

verificándose: $e = a + b + d$, $f = a + c + d$ y $g = b + c + d$.

Así, las columnas $P_1, P_2, P_3, P_4, P_5, P_6, P_7$ son los conjuntos que nos determinan las preguntas que vimos en el demo del juego de la adivinanza, e $y_1, y_2, y_3, y_4, y_5, y_6, y_7$ son las respuestas "$Sí = 1$" o "$No = 0$". Si llamamos $\alpha = \begin{pmatrix} a & b & c & d & e & f & g \end{pmatrix}$ e $y = \begin{pmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 \end{pmatrix}$, pueden suceder dos casos: que sean idénticos o que se diferencien en una sola componente. Equivalentemente, $w(\alpha + y) = 0$ o 1, respectivamente. Si $w(\alpha + y) = 0$, el participante dijo la verdad. Si $w(\alpha + y) = 1$, el participante mintió una vez, y la posición i donde se encuentra el 1 en $\alpha + y$ corresponde a la pregunta P_i, en la cual mintió.

Ejercicio 3: Convéncete que si hay dos errores, corrige mal.

Ejercicio 4: Halla una octava pregunta, que permita hacer lo siguiente: detectar la cantidad de mentiras, si ésta es menor o igual que dos. Si hay una o ninguna mentira se recupera el número. Si hay dos mentiras, se detecta pero no se puede recuperar el número.

1.2 Aplicaciones

La teoría de códigos correctores es una de las aplicaciones más recientes del álgebra. En los años cuarenta del siglo XX, Richard Hamming, uno de los inventores de la teoría de códigos, contaba la siguiente anécdota. Cuando trabajaba para la compañia Bell Laboratories tenía acceso a los ordenadores sólo los fines de semana. Solía dejar corriendo en el ordenador sus programas y cuando

volvía, el fin de semana siguiente, encontraba que alguno de los programas que más necesitaba no habían sido ejecutados (cuando el ordenador detectaba un error en un programa, detenía su realización y pasaba a otro que estaba en la lista de espera). Esto ocasionaba importantes atrasos en su trabajo y le llevó a plantearse el problema de acondicionar de algún modo la información que maneja el ordenador de tal suerte que pudiera corregir los errores.

Imaginemos que se desea enviar información digital (una cadena de ceros y unos) a través de un canal de comunicación de una forma rápida y segura. El canal de comunicación puede ser una línea telefónica, comunicación vía satélite, fibra óptica, almacenamiento de datos en un disco, cinta de computadora, etc. A veces ocurre que el mensaje que se recibe no concuerda con el enviado, principalmente debido a algún error humano, interferencias, deficiencias del equipo, situaciones atmosféricas, etc. Se suele decir que la comunicación se hace a través de un canal con ruido.

Los códigos de Hamming no son tan útiles en la actualidad, pero hay otros como los códigos de Reed-Muller (corrige 5 errores por cada secuencia de 32 bits) o códigos de Golay extremadamente útiles en diversos ámbitos. La transmisión de información desde naves espaciales o a través de satélites de comunicaciones es uno de los paradigmas de la teoría de códigos. Los impresionantes avances tecnológicos, en tecnología digital, que en la actualidad son normales y que consideramos parte de nuestra vida cotidiana, como el teléfono móvil, la televisión digital, los sistemas de navegación aérea y marítima, los CD-R, los DVD, en buena medida (pero no totalmente) no serían posibles sin el desarrollo de los códigos detectores-correctores de error. Estos códigos aparecen, además, en medicina (tomografía), en los códigos de barras, en las transacciones comerciales y bancarias, en sistemas de grabación y reproducción de imágenes, audios y vídeos y, en consecuencia, la importancia de su estudio y de la obtención de resultados originales en este contexto está fuera de discusión.