Miscelánea Matemática **21**, *3-15 (1994)*

Transmisión de imágenes utilizando el código ternario de Golay (una simulación en computadora)

Alida Casale Núñez
Netzahualcóyotl Castañeda Roldán
Martha Salazar Neumann

1 Un poco de historia

Tal vez los códigos más antiguos de la civilización humana sean los códigos de conducta, como los Diez Mandamientos de la ley de Moisés o como el Código de Hammurabi, en los que se prescriben reglas específicas de comportamiento para la convivencia dentro de una sociedad. Con el tiempo, al crecer las sociedades en tamaño y complejidad, también evolucionó la legislación y actualmente tenemos códigos civiles, fiscales, penales y procesales que son tratados especializados muy extensos, detallados y técnicos cuya comprensión constituye una verdadero desafío para el neófito. Sin embargo los expertos dominan muy bien estos códigos refiriéndose a las diversas partes de su contenido mediante números que identifican a cada uno de sus artículos, secciones, capítulos, títulos, etcétera. Estas dos ideas básicas, una la de prescribir procedimientos específicos y la otra la de utilizar un simple número como abreviatura nemotécnica para referirse a algo más largo y complicado, son los puntos de partida para la generalización y evolución del concepto de código.

Como ejemplos sencillos tenemos esas jergas de claves numéricas que utilizan los taxistas, policías y militares en sus comunicaciones radiofónicas. Hay

otro tipo de códigos como el alfabeto de los sordomudos, que utilizan señas con las manos, el de los boy-scouts a base de señales con banderines, y el código de Morse del telégrafo, que representa las letras mediante secuencias de puntos y rayas. En estos códigos observamos que las letras del alfabeto se sustituyen por otros símbolos y que se trata de elegir a los más adecuados en cada situación dada para lograr una transmisión efectiva de la información. Y hablando de transmisión, recordemos que la naturaleza ha diseñado uno de los códigos más asombrosos, complejos y eficaces que conocemos y cuyo propósito es el de transmitir las características fundamentales de cada especie viviente de una generación a otra, sin hacer nunca copias idénticas, sino permitiendo multitud de variaciones individuales dentro del patrón común de cada especie, lo que constituye la base para los procesos de la selección natural.

Así que los códigos resultan indispensables en el mundo que conocemos, no solamente en la naturaleza sino también, y de una forma mucho más notoria, en la tecnología moderna. En la industría de la electrónica, por ejemplo, están los llamados códigos de colores que sirven para identificar los valores de las resistencias y de algunos otros componentes de los circuitos. En nuestros días, la mayor parte de la población urbana está familiarizada con los códigos de barras marcados en las etiquetas y envolturas de los artículos de consumo. Mediante una lectora óptica instalada en cada caja del supermercado, una computadora central está en condiciones de "saber" el precio y demás características del artículo en cuestión. Y aunque a simple vista el código de barras puede parecer algo complicado, recordemos que también hay códigos muy sencillos con un número muy reducido de símbolos, como el semáforo con sus tres luces de colores y su consabido significado.

Hoy por hoy, es en la industría de la computación en donde los códigos encuentran su mayor campo de aplicación y más perspectivas de desarrollo. Muchas veces se dice simplemente "el código del programa" para referirse a un programa que ya está escrito en algún lenguaje de programación en particular. Tenemos el código ASCII, utilizado para representar diferentes caracteres alfanuméricos y funciones simples de impresión como retorno de carro, avance de linea y movimientos de cursor en la pantalla. También están

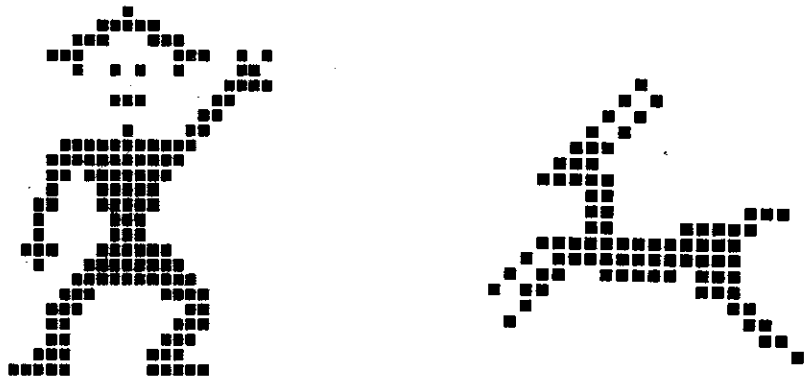

Figura 1: Imágenes originales

los códigos de compactación, que cambian el formato de los datos para poder tener la misma información en un espacio mucho más reducido ahorrando así lugar de almacenamiento en disco.

Aproximadamente desde 1948 se viene estudiando y aplicando la teoría de los *códigos que corrigen errores* en situaciones en las que se plantea el objetivo común de transmitir información proveniente de alguna fuente, a través de algún canal con ruido (interferencia) hasta el receptor, ejemplo de esto son las conversaciones telefónicas, los mensajes telegráficos y recientemente el envío a la tierra de imágenes de otros planetas transmitidas a través de satélites como el Mariner y el Voyager. Para ilustrar este caso supongamos que una nave ha tomado en otro planeta las siguientes fotografías y se dispone a transmitirlas a la tierra: (Figura 1).

La transmisión de estas fotos se lleva a cabo mediante ondas de radio que atraviesan miles de kilómetros en el espacio exterior por lo que están sujetas

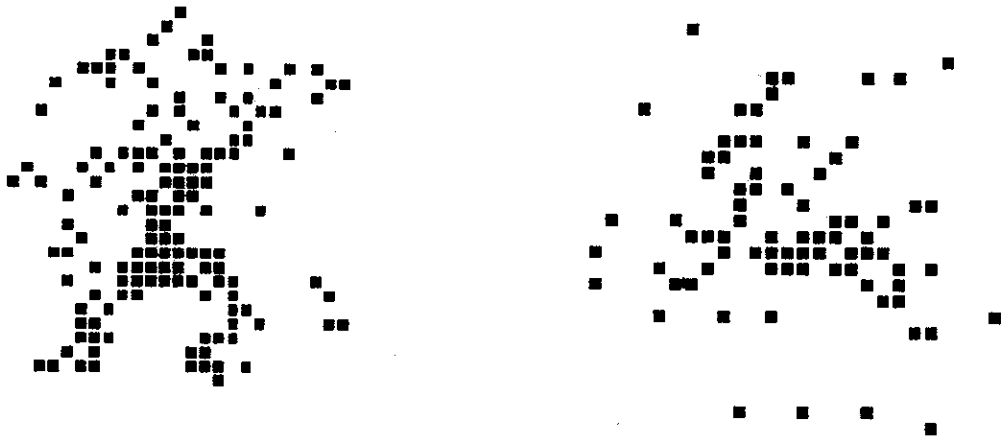

Figura 2: Imágenes recibidas

a diversos tipos de interferencia que distorsionan la señal y provocan errores en las características originales de las que fueron enviadas, de manera que en la tierra se reciben imágenes confusas y borrosas. Si no se contara con un código que detecte y corrija por lo menos algunos de estos errrores no habría forma de saber cuáles puntos de cada imagen recibida pertenecen realmente a la fotografía original y cuáles no. La única información disponible sería algo como esto: (Figura 2).

2 Introducción

El objetivo es mostrar mediante un programa de computadora cómo se utilizan los códigos correctores de errores en un proceso de telecomuniación para recuperar la información perdida durante la transmisión de la señal a causa del ruido y la interferencia presentes en el canal de comunicación.

El programa se desarrolló en lenguaje Turbo Pascal 6.0 y simula la transmisión de un "dibujo" representado como una matriz de puntos de 27×27 que se codifica mediante el código ternario de Golay, un "diccionario" de 279 "palabras" que corrige hasta dos errores en cada secuencia de once dígitos ternarios.

Primero se muestra la imagen recibida después del proceso de transmisión, en el que cada bit de la señal está expuesto de modo aleatorio a ser modificado por la interferencia con una probabilidad llamada "de error" y que se da interactivamente como dato de entrada. Dicha imagen se encuentra distorsionada ya que la modificación de uno cualquiera de los primeros seis bits de cada punto ocasiona que éste cambie de lugar, y sería la única información accesible si no se contara con el código de corrección. Inmediatamente después se exhibe la figura ya corregida, que se obtiene mediante un algoritmo de decodificación basado en el principio de máxima similitud y que identifica correctamente a todos aquellos puntos que fueron transmitidos con un máximo de dos bits erróneos. Aquí se aprecia la eficacia del código para corregir los errores ocurridos, así como la dependencia tan estrecha de tal efectividad con respecto al valor asignado a la probabilidad de error. A continuación aparece el dibujo original y por último se muestran a la vez las tres figuras y una comparación estadística entre la efectividad práctica del código, que depende de cada ejecución, y la teórica, que solamente depende de la probabilidad de error en la transmisión de cada bit.

El programa consta de 24 subrutinas: las primeras cinco se encargan de generar las estructuras algebraicas de datos necesarias para la codificación y decodificación, otras dos se ocupan de cargar desde el disco a la memoria las imágenes con las que se simula la transmisión; una rutina interactiva con el usuario permite que éste elija un dibujo para transmitir y le asigne el valor que desee a la probabilidad de error. Hay otros tres procedimientos que utilizan las estructuras básicas para realizar la simulación propiamente dicha, es decir, la codificación de la figura, la transmisión con interferencia simulada y la decodificación de la señal recibida. Se cuenta también con tres rutinas de monitoreo con las que se puede ver todo el código y las estructuras de decodificación, así como verificar el funcionamiento de la transmisión. Por último,

hay 10 procedimientos de manejo de gráficos en pantalla que son los que permiten la presentación de los dibujos y de las gráficas de probabilidades.

3 Definición de código corrector de errores

Hablaremos únicamente del tipo más simple de códigos correctores de errores: los códigos de bloque, así llamados porque transmiten la información en segmentos o bloques de longitud fija que se pueden decodificar independientemente. En lo sucesivo, al usar la palabra "código" nos estaremos referiendo a este tipo de códigos. En este sentido podemos decir que un *código* es un subconjunto de palabras formadas a partir de un alfabeto determinado. Un código transmite más información de la necesaria, pues si las "palabras" fueran transmitidas tal cuales, el receptor no tendría manera de saber qué "letras" recibió equivocadas.

La idea es trabajar con códigos que tengan alguna estructura algebraica, tomaremos como alfabeto un campo finito F, con $|F| = q$, esto es, con q símbolos; así, si las palabras tienen *longitud* n, estaremos en F^n, que es un espacio vectorial de dimensión n. Definimos, pues, un *código lineal* como un subespacio vectorial de *dimensión* m contenido en F^n.

La distancia entre cualesquiera dos elementos x, y de F^n se determina de la siguiente forma:

$$d(x,y) := |\{i : x_i \neq y_i,\ 1 \leq i \leq n\}|$$

y se conoce como *distancia de Hamming.* Con base en lo anterior, se establece la *distancia mínima* para un código lineal C no trivial, como:

$$\min\{d(x,y) : x, y \in C,\ x \neq y\}$$

Un código $C \subset F^n$ con distancia mínima $2e + 1$ es *perfecto* si para cada $x \in F^n$ existe una única palabra del código $c \in C$ que dista a lo más e unidades de x.

A esta alturas podemos decir que el Código ternario de Golay es un [11,6,5]-código lineal perfecto, esto es, un código de dimensión $m = 6$, subespacio vectorial de un espacio de dimensión $n = 11$, con distancia mínima

5. Para este código $F = Z_3$, es decir, al campo base son los enteros módulo tres.

La forma de obtener el código ternario de Golay que utilizamos para este programa consiste en considerar la factorización del polinomio

$$x^{11} - 1 = (x-1)(x^5 + x^4 - x^3 + x^2 - 1)(x^5 - x^3 + x^2 - x - 1)$$

en polinomios irreducibles sobre el campo de los enteros módulo 3 y considerar también el homomorfismo

$$Z_3[x]/(x^{11} - 1) \to Z_3[x]/(x^5 - x^3 + x^2 - x - 1)$$

definido por dicha factorización. Identificando a cada polinomio con el vector de sus coeficientes tenemos que el núcleo de este homomorfismo es un subespacio vectorial de dimensión 6 contenido dentro del espacio $(Z_3)^{11}$.

Asociadas a esta inmersión particular de $(Z_3)^6$ en $(Z_3)^{11}$ están la matriz generadora y la de chequeo de paridad del código. La primera es de 6×11 y sus renglones son los vectores base del código, el programa utiliza esta matriz para generar a todas las palabras del código como combinaciones lineales de estos seis renglones:

$$\begin{array}{ccccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 2 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 2 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 2 & 2 \\
0 & 0 & 0 & 0 & 1 & 0 & 2 & 1 & 2 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 1 & 2 & 2
\end{array}$$

La matriz de chequeo de paridad es de 5×11 y cada uno de sus renglones consta de los coeficientes de una ecuación que satisfacen todas las palabras del código. Así, esta matriz representa un sistema de ecuaciones que define

al código como su conjunto solución:

1	2	2	2	1	0	1	0	0	0	0
0	1	2	2	2	1	0	1	0	0	0
2	1	2	0	1	2	0	0	1	0	0
1	1	0	1	1	1	0	0	0	1	0
2	2	2	1	0	1	0	0	0	0	1

4 Algunas características del programa

Para simular la transmisión de una imagen, primero consideramos una cuadrícula de 27 renglones por 27 columnas, ya que $27 \times 27 = 729$ es el número de palabras que tiene el código de Golay. Por simplicidad pensamos solamente en este tipo de dibujos cuadriculados y en blanco y negro, es decir, la única información que necesitamos para identificar una figura consiste en saber cuáles cuadros de la cuadrícula pertenecen al dibujo y cuáles no. Ahora numeramos todos los cuadros de manera que al cuadrito de la esquina superior izquierda le toca el número 1 y al de la esquina inferior derecha el 729. Mediante esta numeración identificamos a cada cuadrito con una palabra del código. Así, dada una imagen la pensamos como subconjunto de la cuadrícula y automáticamente tenemos un subconjunto del código. La figura se transmite enviando cada una de las palabras de dicho subconjunto.

El programa no genera los dibujos ni le pide al usuario que los haga durante el tiempo de ejecución, sino que supone que ya están hechos y los busca en el disco dentro de un archivo de texto previamente generado por un programa de edición. En este archivo se encuentran guardadas las figuras en forma de listas de números correspondientes a los cuadritos del dibujo. Las palabras del código que se van a transmitir son precisamente las que tengan esos números.

Otro supuesto básico del programa es que el nivel de interferencia permanezca constante durante todo el tiempo que dura la transmisión de una imagen. Antes de iniciar con este proceso se le debe asignar un valor a la variable p que representa la probabilidad de que la interferencia modifique

una coordenada de la señal. Cada palabra del código (una señal) es una secuencia de once dígitos ternarios y cada uno de éstos se transmite independientemente. La transmisión de cada coordenada se simula utilizando la función "Random" para decidir si el dígito en esa coordenada en particular se respeta ó se modifica dándole otro valor distinto del original. Es como si jugáramos una serie de once volados con una moneda cargada de modo tal que nuestra probabilidad de ganar es $1-p$ y nuestra probabilidad de perder es p. Ganar el volado significa respetar el dígito en cuestión y transmitir correctamente esa coordenada, mientras que perderlo significa introducir un error y cambiar el valor de la coordenada original. Después de haberse jugado esta serie de once volados la palabra del código puede haber salido ilesa ó bien puede haber sufrido modificaciones, aunque cuando esto sucede todavía le quedan esperanzas de recuperación, ya que la estructura algebraica del código es capaz de componer la palabra averiada siempre y cuando ésta no traiga más de dos golpes, es decir, dos coordenadas erróneas. Si en el proceso de transmisión la palabra sufrió tres modificaciones o más, ya puede irse olvidando de su verdadera identidad pues ya no habrá quién pueda reconocerla, en este caso se le confundirá irremediablemente con alguna otra palabra del código.

5 Algoritmo de decodificación

Al recibir la señal de once dígitos ternarios se determina fácilmente si ésta pertenece ó no al código utilizando la matriz de chequeo de paridad. Si al sustituir nuestro vector recibido en cada una de las cinco ecuaciones obtenemos como resultado 0, entonces se trata de un elemento del código. Esta es una condición necesaria y suficiente ya que el código es el subespacio vectorial formado por las soluciones del sistema de ecuaciones. Si alguna de las ecuaciones no se cumple, sabremos que la señal recibida contiene al menos un error.

Pero en realidad la matriz pone más herramientas a nuestro alcance ya que podemos utilizar a las ecuaciones no como tales sino como funciones

lineales de once variables. Al tener cinco ecuaciones tenemos cinco funciones y al agrupar a éstas en un vector tenemos una transformación lineal T que va de $(Z_3)^{11}$ en $(Z_3)^5$. Esta transformación lineal tiene dos propiedades muy importantes que son:

1. Restringida a la bola de radio 2 con centro en el origen es biyectiva.

2. Es periódica con cualquier elemento del código como periodo.

Estas propiedades son importantes porque el código de Golay es un código perfecto desde el que se puede cubrir a todo el espacio $(Z_3)^{11}$ con bolas de radio 2 centradas en los puntos del código y esta cubierta no es nada más una cubierta sino una partición del espacio en clases de equivalencia ajenas entre sí. De manera que al recibir una señal v simplemente la sustituímos en los primeros miembros de las ecuaciones y obtenemos $T(v)$, su imagen ó "sindrome", un vector de cinco coordenadas que tiene una única preimagen dentro de la bola de radio dos alrededor del origen en $(Z_3)^{11}$. Si suponemos que la señal recibida no contiene más de dos errores (lo que resulta muy probable para valores razonables de la probabilidad de error p), entonces la señal recibida v se encuentra a lo más a 2 unidades de distancia del punto del código c del que proviene, y tenemos que $v = c + e$, en donde e es el vector de los errores cometidos, en este caso más probable un vector con menos de tres coordenadas diferentes de cero. Por tanto tenemos

$$T(v) = T(c+e) = T(c) + T(e) = 0 + T(e) = T(e)$$

y esta igualdad distingue de manera inequívoca al vector e puesto que dentro de la bola de radio 2 alrededor del origen sólo hay un vector cuya imagen bajo T coincida con la de v. Así es que una vez identificado el error e podemos recuperar al elemento original c del código restando

$$c = v - e$$

Para poder identificar rápidamente los errores, el programa genera a todos los vectores e de la bola de radio dos con centro en el origen de $(Z_3)^{11}$ y para

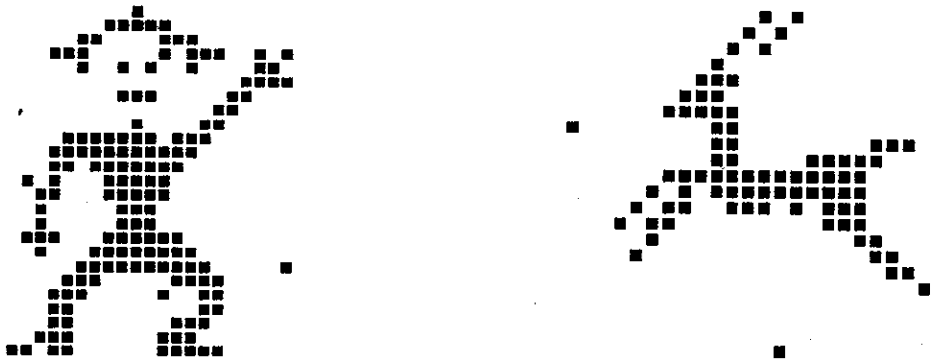

Figura 3: Imágenes decodificadas

cada uno de ellos calcula su imagen $T(e) \in (Z_3)^5$ y la interpreta como número entero mediante la notación posicional de base 3 obteniendo así una función biyectiva entre todos los errores corregibles posibles y los números enteros del 0 al 242. Después, al recibir una señal cualquiera v se usa la inversa de esta función biyectiva para saber cuál es el error e que tiene el mismo síndrome $T(v)$ y ése es el error que se corrige.

Después de decodificar cada una de las palabras recibidads en la transmisión, se vuelven a interpretar estas palabras corregidas como cuadritos del dibujo y ahora la apariencia de la imagen queda muy mejorada como se puede apreciar a continuación: (Figura 3).

6 Efectividad del código

Podemos definir la efectividad del código para transmitir este tipo particular de imágenes que estamos manejando, en términos de funciones de probabilidades de "bit malo–punto bueno". Se trata de medir hasta qué punto el código consigue nuestro objetivo de proporcionar a la información una

estructura cuya solidez global soporte las fallas de los componentes individuales. Cada punto del dibujo está identificado con una palabra del código que es un vector de once bits ó coordenadas, cada una de las cuales está sujeta a ser modificada con una probabilidad p. ¿Cuál es la probabilidad $g(p)$ de que dado un punto cualquiera del dibujo, después de atravesar por la modificación aleatoria sea interpretada correctamente por nuestro sistema de decodificación? Esta es la pregunta clave.

La transmisión de cada coordenada es independiente de las de todas las demás, por lo que la probabilidad de que ocurran exactamente k errores en el vector del código correspondiente a un punto del dibujo es

$$\binom{11}{k} p^k (1-p)^{11-k}$$

Como el código de Golay corrige solamente hasta dos coordenadas erróneas por vector, la probabilidad de decodificar correctamente el punto recibido está dada por

$$\sum_{k=0}^{2} \binom{11}{k} p^k (1-p)^{11-k} = (1-p)^9 (1+9p+45p^2)$$

Debemos comparar esta función de efectividad con la que corresponde a la transmisión de la información desnuda, es decir sin ninguna codificación. En este caso los vectores de dígitos ternarios ya no son de longitud once sino solamente de seis que es la dimensión del código. Pero aquí cualquier modificación aunque sea de una sola de las coordenadas implica perder el punto ya que no hay posibilidades de decodificación, así que para interpretar un punto correctamente después de la transmisión ésta debe de ser perfecta, sin ninguna coordenada errónea. En estas condiciones la probabilidad $f(p)$ de transmitir un punto correctamente es:

$$f(p) = (1-p)^6$$

En la figura siguiente mostramos una comparación gráfica entre estas dos funciones: (Figura 4).

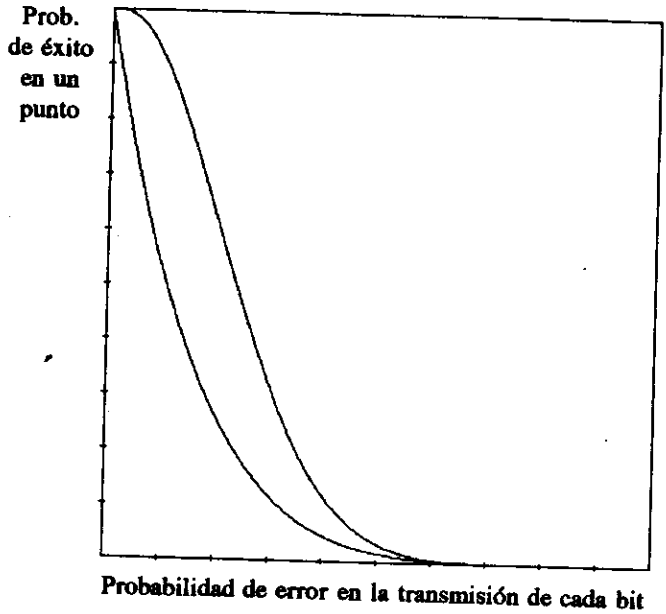

Figura 4: Probabilidades: Decodificación correcta contra error

Bibliografía

[1] *Introduction to coding theory* J. H. Van Lint. Springer Verlag, 1982.

[2] *Matemáticas en las ciencias del comportamiento* R.Carnap, O. Morgenstern, N. Wiener y otros. Alianza Editorial (AU 86). pags. 119–135.

[3] *Matemáticas discreta y combinatoria* Ralph P. Grimaldi. Addison–Wesley Iberoamericana. pags. 334–353.

[4] *A first course in coding theory* Raymond Hill. Oxford (Applied mathematics and computing science series).