Una introducción a los códigos lineales

Francisco Galluccio

11 de diciembre de 2025

1. Introducción

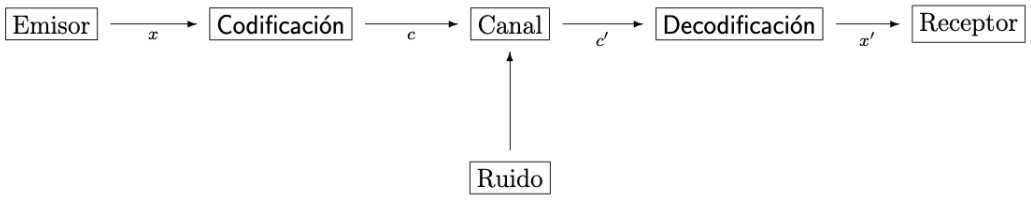
Supongamos que estamos de vacaciones en Hurlingham y queremos mandarle una foto a nuestra familia. Claramente podríamos imprimir la foto que nos hemos sacado, ir al correo y enviar por ese medio una foto... pero esa opción es mucho más costosa y complicada que apretar dos botones de nuestro teléfono y enviarla por nuestra red social favorita. ¡Además, podría dañarse la foto al imprimirse o transportarse!

¿Cómo hace entonces esa red social para transportar eficiente y fehacientemente nuestra foto por medio del internet?

1.1. El problema

El objetivo de la teoría de códigos es conseguir transmitir información de manera eficiente y segura desde un emisor hacia un receptor, por medio de un canal.

La situación general es, *a grosso modo*, la siguiente. Supongamos que queremos enviar un mensaje. Este es enviado por un canal de comunicación, cuyas características dependen de la naturaleza del mensaje a ser enviado (i.e. sonido, imagen, datos). En general, hay que hacer una traducción entre el mensaje original (o palabra fuente) x y el tipo de mensaje c que el canal está capacitado para enviar (palabras código). Este proceso se llama codificación. Una vez codificado el mensaje lo enviamos a través del canal, y nuestro intermediario (el receptor) recibe un mensaje codificado (palabra recibida) posiblemente erróneo, ya que en todo proceso de comunicación hay ruido e interferencias. El mensaje recibido c' es traducido nuevamente a términos originales x', es decir, es decodificado. Todo el proceso se resume en el siguiente esquema:



Vayamos a la codificación:

Definición 1.1. Un **alfabeto** A es un conjunto finito de caracteres. Una **palabra** es una sucesión de caracteres de este alfabeto, y un **mensaje** es una sucesión de palabras.

El conjunto de palabras de longitud n se denota por A^n.

Ejemplo 1.2.

1. El alfabeto usual $A = \{a, b, \ldots, x, y, z\}$. Cualquier sucesión de caracteres es una palabra.
2. Los bits $B = \{0, 1\}$. Las palabras son los *bytes*, cadenas de 8 bits.
3. Los números del 1 al 10. Las palabras son cadenas de números.

Definición 1.3. Supongamos que A posee q caracteres. Un (n, k) código C en A es un subconjunto de A^n, de q^k palabras de longitud n. Los elementos de C se llaman **palabra código**.

Una **codificación** del código es una función E inyectiva de A^k hacia A^n. La **tasa** de transmisión de un (n, k) código C es $R = \dfrac{k}{n}$.

Observemos que luego de codificar, se necesitan más caracteres para transmitir la misma cantidad de información. En efecto, para transmitir un mensaje de B caracteres utilizando un código, se necesitarían enviar al menos $\frac{B}{k} \cdot n = \frac{B}{R} = B'$ caracteres. Notemos que $0 \leq R \leq 1$, y los casos en los que alcanza los extremos son casos triviales.

Ejemplo 1.4. Para el alfabeto $A = \{0, 1\} = \mathbb{Z}_2$ tenemos los siguientes códigos y sus codificaciones:

1. El $(3, 2)$ código de repetición $C_1 = \{000, 111\}$ con codificación trivial.
2. El $(8, 7)$ código de paridad C_2 con codificación $E(x) = (x \mid b)$, donde $b = 1$ si hay una cantidad impar de 1 en x y $b = 0$ si no. Hay $2^7 = 128$ palabras en C
3. El $(6, 3)$ código de verificación C_3 con codificación $E(abc) = (abcxyz)$ donde

$$x = 1 \text{ si } a \neq b, \qquad y = 1 \text{ si } a \neq c, \qquad z = 1 \text{ si } b \neq c.$$

Ejercicio 1.5. Determinar una condición necesaria y suficiente para que una palabra $x = (x_1, x_2, \ldots, x_8)$ pertenezca al código C_2 del ejemplo 1.4.

Ejercicio 1.6. Escribir todas las $2^3 = 8$ palabras del código C_3

Definición 1.7. Supongamos que se codificó una palabra código $x \in C$. Decimos que se cometió un **error** en la transmisión si se recibe una palabra $y \in A^n$ distinta de x en alguna coordenada. Decimos que se **detectó** un error si la palabra recibida y no está en C.

Cada uno de los códigos del ejemplo puede detectar **hasta** un error.

Definición 1.8. Sean $u = (u_1, u_2, \ldots, u_n)$ y $v = (v_1, v_2, \ldots, v_n)$ palabras de A^n. Decimos que u, v difieren en la **posición** i ó en la coordenada i si $u_i \neq v_i$. La **distancia de Hamming** entre u y v es la cantidad de posiciones en las que difieren, y se denota por $d(u, v)$.

Ejercicio 1.9. Muestra que para todos $u, v, w \in A^n$ se verifica

- $0 \leq d(u, v) \leq n$,
- $d(u, v) = d(v, u)$,
- $0 = d(u, v) \iff u = v$,
- $d(u, v) \leq d(u, w) + d(w, v)$,
- $d(u, v) = d(u + w, v + w)$. si A es un espacio vectorial.

Definición 1.10. Sea C un (n, k) código sobre A. La **distancia mínima** $d = d(C)$ del código es

$$d = \text{mín}\{d(u, v) : \ u, v \in C, \ u \neq v\}.$$

Con $0 < d \leq n$ decimos que C es un (n, k, d) código.

Definición 1.11. Supongamos que se codificó una palabra código $x \in C$. Decimos que se **cometió** un **error de peso** t en la transmisión si se recibe una palabra $y \in A^n$ con $d(x, y) = t$. Decimos que se **detectó** un **error de peso** t si la palabra recibida y no está en C. Decimos que es posible **detectar hasta** t errores si siempre que $d(x, y) \leq t$ se detecta un error. Decimos que es posible **corregir hasta** t errores si, siempre que $d(x, y) \leq t$, se puede recuperar el mensaje original x al haber recibido y a sabiendas que se detectó un error de peso t.

Proposición 1.12. *Se pueden detectar hasta t errores si y sólo si $d(C) \geq t+1$. Se pueden corregir hasta t errores si y sólo si $d \geq 2t+1$.*

Demostración. Supongamos que para una palabra código x enviada y toda palabra w recibida, con $d(x,w) \geq t$, resulta que w no puede ser una palabra código, por lo que se detecta la existencia del error. De este modo $d \geq t+1$ es una condición necesaria y suficiente. Supongamos ahora que hay 2 palabras código x, v de C a distancia menor ó igual a $2t$. Luego existe una palabra w que está a distancia menor ó igual a t de cada una de x y v. Si se envía x ó v y se recibe la palabra w en lugar de ellas, entonces en ambos casos hubo a lo más t errores y no es posible determinar cuál ha sido el mensaje original. De este modo vemos que para corregir t errores es necesario que $d \geq 2t+1$. Recíprocamente, si $d \geq 2t+1$ y se recibe una palabra w junto a un error de peso t, vemos que debe existir al menos una palabra código x con $d(x,w) \leq t$. Si existiera otra palabra código v a distancia menor ó igual a t, se obtiene $d(u,v) \leq d(u,w) + d(w,v) = 2t < d$, absurdo. □

Observación: ¿Cómo sabemos que se cometieron t errores, al recibir w, para poder corregirlos, y no se cometieron, en realidad $d-t$ errores?. ¡No lo sabemos!. Pero para ello podemos buscar d suficientemente grande bajo la siguiente premisa:

Supongamos que nuestro canal tiene una **probabilidad de error** simétrica, es decir, que cada coordenada tiene la misma probabilidad p de cambiarse por cualquier otro símbolo del alfabeto, donde p sólo depende del canal, y estos errores son independientes. Luego

Proposición 1.13. *La probabilidad de que haya al menos un error es*

$$P(Error) = 1 - (1-p)^n,$$

la probabilidad de que haya un error de peso t es

$$P(t\ Errores) = \binom{n}{t} p^t (1-p)^{n-t},$$

y por lo tanto la probabilidad de que haya algún error de peso a lo más t es

$$\sum_{k=0}^{t} \binom{n}{k} p^k (1-p)^{n-k}.$$

Demostración. La probabilidad de que ningún error ocurra en la palabra es igual al

producto sobre todas las coordenadas de que ningún error ocurra, lo cual es $(1-p)^n$. Luego la probabilidad de que al menos un error ocurra es $1-(1-p)^n$. □

Teorema 1.14. *La probabilidad P de que una palabra recibida se decodifique correctamente es igual a la probabilidad de que haya a lo más t errores en ella, donde $t = \lfloor \frac{d-1}{2} \rfloor$. La probabilidad de que un mensaje decodifique correctamente es igual a P^ℓ donde ℓ es la cantidad de palabras enviadas.*

Ejercicio 1.15. Suponga que se envía un mensaje de longitud $10^5 = 10000$ sobre un canal con probabilidad $p = 10^{-3} = 0,001$ de error. Muestra que la probabilidad de decodificar correctamente el mensaje con el código C_1 del ejemplo 1.4 es aproximadamente $0,970$; la probabilidad de decodificar correctamente el mensaje con el código C_2 del ejemplo es aproximadamente $0,000011$; y que la probabilidad de decodificar correctamente el mensaje con el código C_3 del ejemplo es aproximadamente $0,951$.

2. Códigos Lineales

Volvamos ahora a los códigos. Para un (n,k,d) código no hicimos ninguna observación sobre cómo o cuáles debían ser las q^k palabras de éste.

Definición 2.1. Sea p un número primo. El conjunto de los restos $\{0,1,\ldots,p-1\}$ en la división por p forman un cuerpo $\mathbb{F}_p$.

Ejercicio 2.2. Muestra que $\mathbb{F}_p$ es un cuerpo. Esto es, que se puede sumar, restar, multiplicar, dividir, y verifica los axiomas de conmutatividad, asociatividad, distributiva, existencia de inversos y de elemento neutro, donde la operación es la habitual. Concluye que $\mathbb{F}_p^n$ es un espacio vectorial de dimensión n.

Definición 2.3. Supongamos entonces que $A = \mathbb{F}_q$ donde q es primo. Un (n,k,d) **código lineal** $C \subset \mathbb{F}_q^n$ es un subespacio vectorial de $\mathbb{F}_q^n$ de dimensión k. Si $q = 2$ entonces C se llama **código binario**.

Ejemplo 2.4. El código C_3 del ejemplo 1.4 es un subespacio vectorial de $\mathbb{F}_2^6$. Tiene dimensión 3 y está generado por los tres vectores

$$\{(1,0,0,1,1,0),\ (0,1,0,1,0,1),\ (0,0,1,0,1,1)\}.$$

Las restantes $8-3=5$ palabras son la palabra nula, las 3 sumas de 2 de ellas, y la suma de las 3.

Proposición 2.5. *Si C es un (n, k, d) código lineal, entonces la distancia mínima d del código es igual al* ***peso mínimo***

$$w = \text{mín}\{w(c) : \ c \in C, \ c \ \neq 0\},$$

donde $w(c)$ cuenta la cantidad de coordenadas no nulas de la palabra c.

Ejercicio 2.6. Muestra que la distancia mínima del código C_3 es igual a 3.

Ejercicio 2.7. Halla la distancia mínima del código C_2. Exhibe un par de palabras con esa distancia y justifica por qué no puede haber otro par con distancia menor.

Notemos entonces que, para un n fijo, si la dimensión k de un código es muy grande, entonces es posible que haya dos palabras muy parecidas, de modo que la distancia d sea relativamente pequeña. La pregunta más importante que puede surgir ahora es qué tan grande podemos hacer, simultáneamente, la dimensión y la distancia.

Teorema 2.8 (Cota de Singleton)**.** *Sea C un (n, k, d) código sobre $\mathbb{F}_q$. Entonces se verifica que*

$$k + d \leq n + 1.$$

Demostración. Sea $W = \{(x_1, x_2, \ldots, x_n) \in \mathbb{F}_q^n : \ x_d = x_{d+1} = \ldots = x_n = 0\}$ el cual es un subespacio vectorial de dimensión $d - 1$. Notemos que $W \cap C = \{\vec{0}\}$, el vector nulo. En efecto, si $c \in W \cap C$ debe ser $c_d = c_{d+1} = \ldots c_n = 0$ por lo que $d(c, \vec{0}) \leq d - 1$, y por lo tanto como la distancia mínima era d, debe ser $c = \vec{0}$.

Luego W y C son subespacios de $\mathbb{F}_q^n$ con intersección trivial, y viendo dimensiones obtenemos $(d - 1) + k \leq n$ de donde $d + k \leq n + 1$. □

Una observación muy importante es que la cota de Singleton es válida aún en los casos de los códigos no lineales. Casos donde se alcanza la cota, es decir para $|C| = q^{n+1-d}$, se llaman códigos **MDS**.

Ejercicio 2.9. Muestra la cota de Singleton aún en el caso de que C no sea un código lineal. (*sugerencia: ¿qué sucede si se eliminan las primeras d coordenadas del código?*)

Esta idea de trabajar con coordenadas y algunos vectores en particular nos permite construir lo siguiente

Definición 2.10. Sea C un (n, k, d) código lineal sobre $\mathbb{F}_q$, y supongamos que $c_1, c_2, \ldots, c_k$ son una base de él como espacio vectorial de $\mathbb{F}_q^n$. Luego $\{c_1, \ldots, c_k\}$ son un **conjunto de generadores**. Una matriz G de $k \times n$ cuyas *filas* son un conjunto de generadores de C se llama **matriz generadora**.

Podemos ver que la codificación de un mensaje embebido en $\mathbb{F}_q$ puede entenderse como la multiplicación de un vector $m^T \in \mathbb{F}_q^k$ por la matriz G.

Ejercicio 2.11. Una matriz G de $k \times n$ es una matriz generadora (de algún código) si y sólo si es de rango k.

Ejemplo 2.12. El $(7, 4, 3)$ código de Hamming generado por los vectores

$$\{(1,0,0,0,1,1,0), (0,1,0,0,1,0,1), (0,0,1,0,0,1,1), (0,0,0,1,1,1,1)\},$$

posee matriz generadora

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Antes nos preguntábamos por una condición necesaria y suficiente para que una palabra w dada pertenezca a un código. Usando que ahora estamos en un subespacio vectorial, podemos pensar si la palabra w verifica las ecuaciones lineales que definen al subespacio.

Definición 2.13. Sea C un (n, k, d) código lineal sobre $\mathbb{F}_q$. Una matriz H de $m \times n$ se llama **matriz de paridad** (ó matriz de **control de paridad**, o simplemente matriz de **control**) si $Hc^T = 0$ para toda $c \in C$. Equivalentemente, si HG^T es una matriz nula.

Ejemplo 2.14. Para el código C_4 del ejemplo 2.12 vemos que

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

Teorema 2.15. *Supongamos que C es un (n, k, d) código lineal sobre $\mathbb{F}_q$, que A es una matriz de $k \times (n-k)$ tal que $G = (I_k \mid A)$ es una matriz generadora de C (esto es, que G está en **forma estándar**). Luego $H = (-A^T \mid I_m)$ es una matriz de*

paridad, con $m = n - k$. Más aún, son las únicas matrices tales que sus primeras (últimas) columnas forman la matriz identidad.

Corolario 2.16. *Cualquier matriz de control de paridad tiene rango $n - k$*

Con una matriz de control de paridad podemos verificar inmediatamente si una palabra $w \in \mathbb{F}_q^n$ es una palabra código, al verificar si $Hw^T = 0$.

Pero el código de Hamming C_4 posee distancia mínima 3. ¿Y si queremos corregir un error?

Proposición 2.17. *Supongamos que la matriz H no posee ninguna columna con ceros, ni dos columnas linealmente dependientes. Luego es posible corregir al menos un error.*

Demostración. Mostraremos que la distancia mínima es $d \geq 3$. Equivalentemente, que el peso mínimo es $w \geq 3$. Como no hay columnas con ceros, entonces no puede existir palabra $c \in C$ con $w(c) = 1$ ya que en caso contrario para $c = (0, 0, \ldots, c_i, \ldots, 0, 0)$ la condición $Hc^T = 0$ implica que la columna i, H_i de H es nula. De manera análoga, si $w(c) = 2$ resultaría $c_i, c_j \neq 0$ por lo que $c_i H_i + c_j H_j = 0$ implica que las columnas i, j de H son linealmente dependientes. □

Ejercicio 2.18. Sea H una matriz de paridad de un (n, k, d) código lineal C. Halla mín$\{r :$ hay r columnas linealmente dependientes en $H\}$ en función de n, k, d.

En particular podemos definir

Definición 2.19. Sea H una matriz de r filas y n columnas, con componentes en $\mathbb{F}_2$. Luego hay a lo más $2^r - 1$ posibles columnas para H. Si $n = 2^r - 1$, vemos que $d = 3$ por la proposición anterior y $k = n - r = 2^r - 1 - r$ en virtud del corolario 2.16, el código $H(r)$ con matriz de paridad H se llama **código de Hamming**.

La importancia del código $H(3) = C_4$ radica en que su matriz de control de paridad posee el máximo número de columnas (es decir están todos los vectores no nulos de $\mathbb{F}_2^3$ por lo que la longitud de $H(3)$ es la máxima posible dentro de todos los códigos cuya matriz de paridad es de rango r.

Ejercicio 2.20. Generalizar la construcción de **código de Hamming** para $\mathbb{F}_q$, denotándolo por $H_q(r)$. Mostrar que $n = \frac{q^r - 1}{q - 1}$, $k = n - r$ y $d = 3$.

En general, con cualquier código lineal C con $d = 3$, podemos detectar a lo más 2 errores, y ahora mostraremos que es posible corregir un único error sin necesidad de buscar las 2^k palabras de C:

Supongamos que recibimos la palabra $c' = c + e$ donde $c \in C$ es la palabra código enviada y e es un error. Como el error es de peso 1, entonces $e = e_i$ es un vector canónico de $\mathbb{F}_2^n$, y obtenemos

$$Hc'^T = H(c + e_i)^T = Hc^T + He_i^T = He_i^T = H_i,$$

por lo que buscando las n columnas de H, podemos determinar que el error se cometió en el $i-$ésimo componente de c.

¿Y si queremos corregir más de un error? Sabemos que Hc'^t es un vector no nulo el cuál sólo depende del error e. Por lo tanto, en lugar de determinar la palabra código enviada, podemos buscar el error cometido y recuperar $c = c' - e$. Acá es donde usamos la particularidad de que el peso del error es a lo sumo t: podríamos buscar sólamente entre las palabras de peso menor ó igual a t, en vez de las 2^k palabras del código.

Definición 2.21. Supongamos que trabajamos con un código lineal sobre $\mathbb{F}_q$. Supongamos que se recibe una palabra c'. Decimos que el **síndrome** de c' es $s(c') = Hc'^T$. Un **arreglo estándar** para el código C es un arreglo o tabla de $q^{n-k} \times q^k$ donde la primer fila consiste en las palabras de C y la primer columna consiste en palabras de $\mathbb{F}_q^n$ de peso menor ó igual a t ordenadas crecientemente por peso; mientras que en la posición i, j de la tabla se obtiene la suma (en $\mathbb{F}_q^n$) de su líder de fila y columna.

Ejemplo 2.22. Supongamos que tenemos el código C_3, con $n = 6$ y $k = 3$. Luego tenemos que armar una tabla de $2^3 \times 2^3$. En la primer fila ponemos las palabras código, comenzando por el vector 0.

000000	100110	010101	001011	110011	101010	011110	111000

Luego añadimos alguna de las palabras de peso 1 que aún no haya aparecido en el arreglo,

000000	100110	010101	001011	110011	101010	011110	111000
100000	000110	110101	101011	010011	001010	111110	011000

Seguimos añadiendo las palabras de peso 1 (hasta $t \leq (d-1)/2$) hasta que estén

todas. En este caso $t = 1$ y

000000	100110	010101	001011	110011	101101	011110	111000
100000	000110	110101	101011	010011	001101	111110	011000
010000	110110	000101	011011	100011	111101	001110	101000
001000	101110	011101	000011	111011	100101	010110	110000
000100	100010	010001	001111	110111	101001	011010	111100
000010	100100	010111	001001	110001	101111	011100	111010
000001	100111	010100	001010	110010	101100	011111	111001

Hasta aquí podemos detectar y corregir el error. Ahora completamos con las palabras de peso entre t y d que aún no aparezcan. En este caso falta sólamente 100001, la cual está a distancia mayor a 1 de toda palabra del código. Tenemos la fila

100001	000111	110100	101010	010010	001100	111111	011001

Teorema 2.23. *Supongamos que trabajamos con un (n, k, d) código lineal sobre $\mathbb{F}_q$. Supongamos que se recibe una palabra c', y que el error es de peso menor ó igual a t. Luego el error cometido e es el único vector de peso menor ó igual a t, entre los q^{n-k} líderes de fila, tal que su síndrome $s(e)$ es igual al síndrome $s(c')$. La palabra enviada es $c = c' - e$ que es el líder de su columna, y en particular e era el líder de su fila. Si la palabra recibida no comparte síndrome con las primeras filas, entonces se detectaron al menos $t + 1$ errores y no es posible corregirlos.*

¿Y si queremos códigos con más estructura? En este caso queremos buscar ejemplos de códigos que verifiquen alguna de las siguiente:

1. Tengan distancia $d \geq 5$, aunque manteniendo dimensión cercana a su longitud,
2. Tengan un proceso de corrección mucho más eficiente,
3. Tengan un proceso de decodificación mucho más eficiente.

Para el primero de estos problemas, podemos pensar en lo siguiente:

Definición 2.24. Un (n, k, d) código lineal se dice **perfecto** si para cada palabra $v \in \mathbb{F}_q^n$ existe una única palabra código $c \in C$ con $d(c, v) \leq t$, con $d = 2t + 1$. Geométricamente, si las bolas de radio r centradas en las palabras código cubren, sin huecos ni superposiciones, el espacio $\mathbb{F}_q^n$.

Ejercicio 2.25. Muestra que ser perfecto es equivalente a

$$q^{n-k} = \sum_{i=0}^{t} (q-1)^i \binom{n}{i},$$

Ejercicio 2.26. Muestra que los códigos de hamming $H_q(r)$ son perfectos con $d = 3$.

Ejercicio 2.27. Muestra que ser perfecto es equivalente a que la tabla de síndromes tenga como líder de fila todas las palabras de peso $\leq t$.

El segundo de los distintos de estos códigos en aparecer fue el **código de Golay**, con parámetros $(n, k, d) = (24, 12, 8)$, sobre $\mathbb{F}_2$. No es trivial que $n = 23, t = 3$ sea una solución no trivial a la ecuación $\sum_{i=0}^{t} \binom{n}{i} = 2^m$, pero ahora es conocido que el código es el segundo y anteúltimo de los pocos de los distintos.

Éste código puede construirse mediante el código de Hamming o mediante otras ideas geométricas. Los siguientes son ejemplos

Ejemplo 2.28. El $(24, 12, 8)$ código lineal $\mathcal{G}_{24}$ se obtiene con matriz generadora:

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

mientras que el código de Golay $\mathcal{G}_{23}$ se obtiene de G eliminando la última columna. Es posible mostrar que $\mathcal{G}_{23}$ es un $(23, 12, 7)$ código perfecto.

Comparando el código de Golay $\mathcal{G}_{23}$ con el Código de Hamming $H(3)$, vemos que los parámetros (n, k, d) son comparables en el siguiente sentido:

- Para $H(3)$ tenemos $n = 7,\ k = 4,\ d = 3$. Luego $R = \frac{k}{n} = \frac{4}{7}$ y $\delta = \frac{d}{n} = \frac{3}{7}$.

- Para $\mathcal{G}_{23}$ tenemos $n = 23,\ k = 12,\ d = 7$. Luego $R = \frac{k}{n} = \dfrac{12}{23}$ y $\delta = \frac{d}{n} = \dfrac{7}{23}$.

Si bien los **parámetros relativos** son *peores*, ahora podemos corregir 3 errores (en lugar de uno), y la estructura simétrica del código de Golay permite hacer la corrección mucho más eficiente. Además, ahora tenemos $2^{12} = 4096$ palabras código, comparadas con las $2^4 = 16$ de Hamming!

Ejemplo 2.29. Si quisiesemos transmitir una foto, donde cada pixel está asociado a una terna (R, G, B) para $0 \leq R, G, B \leq 255 = 2^8 - 1$ necesitaríamos entonces $256^3 = 2^{24}$ palabras código por lo menos!

Ejercicio 2.30. ¿Usar dos copias del código $\mathcal{G}_{24}$ es *mejor* que juntar 6 copias de $H(3)$?

3. Códigos Cíclicos y otros códigos

Definición 3.1. Sea C un (n, k, d) código lineal sobre $\mathbb{F}_q$. Decimos que C es un **código cíclico** si, siempre que $c = (c_1, c_2, \ldots, c_n) \in C$, resulta que la permutación $\sigma(c) = (c_n, c_1, c_2 \ldots, c_{n-1})$ también está en C.

Ejemplo 3.2. Reordenando convenientemente las coordenadas, los códigos de Golay $\mathcal{G}_{23}$ y $\mathcal{G}_{11}$ son cíclicos. Del mismo modo el código de Hamming $H(3)$ es cíclico. El código C_3 no lo es.

Cambiando los índices y haciendo una biyección entre $\mathbb{F}_q^n$ y $\mathbb{F}_q[x]$ dada por

$$\phi(c_0, c_1, \ldots, c_{n-1}) = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1},$$

podemos estudiar a C como un subespacio de $\mathbb{F}_q[x]$. Si queremos ver la propiedad de ser cíclico, podemos pensar primero que $\phi(\sigma(c)) = c_n + c_0 x + \ldots + c_{n-2} x^{n-1}$ lo cual es muy similar a $x\phi(c)$, por lo que si $x^n = 1$ resultaría $\phi(\sigma(c)) = x\phi(c)$.

Teorema 3.3. *Sea C un (n, k, d) código lineal. Luego C es cíclico si y sólo si $\phi(C) \subset \dfrac{\mathbb{F}_q[x]}{\langle x^n - 1 \rangle}$ es un ideal.*

Corolario 3.4. *Sea C un (n, k, d) código cíclico. Luego existe un polinomio irreducible $g(x) \in \mathbb{F}_q[x]/(x^n - 1)$ de grado $n - k$ tal que $\phi(C) = (g(x))$, es decir, es el ideal generado por $g(x)$.*

En efecto, podemos encontrar $g(x) = c_0+c_1x+c_2x^2+\ldots+c_{n-k}x^{n-k}$, con $c_0, c_{n-k} \neq 0$, de modo que la matriz generadora de C es

$$\begin{pmatrix} c_0 & c_1 & c_2 & \cdots & c_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & c_0 & c_1 & \cdots & c_{n-k-1} & c_{n-k} & 0 & \cdots & 0 \\ 0 & 0 & c_0 & \cdots & c_{n-k-2} & c_{n-k-1} & c_{n-k} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & c_0 & c_1 & c_2 & \cdots & c_{n-k} \end{pmatrix}$$

Ejercicio 3.5. Muestra que si C es cíclico, entonces

$$\phi(C) = \{p(x)g(x) : \deg(p) < k\}.$$

Ejercicio 3.6. Muestra que si C es cíclico, entonces las últimas k coordenadas de cada palabra determinan unívocamente las $n - k$ restantes.

Los códigos cíclicos tienen un proceso de codificación, decodificación, detección y corrección de errores sencillo de implementar cuando se conoce la factorización de $x^n - 1$ en $\mathbb{F}_q[x]$.

Proposición 3.7. *Supongamos que $x^n - 1 = g(x)h(x)$. A cada $c = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ le asociamos $\phi(c) = \sum_{i=0}^{n-1} c_i x^i$. Son equivalentes*

- $c \in C$
- $g(x) \mid \phi(c)$,
- $x^n - 1 \mid \phi(c)h(x)$,
- $\phi(c)h(x) = 0$ *en* $\mathbb{F}_q^n[x]/(x^n - 1)$.

Esta noción de pensar a una palabra como a una función, puede ser usado como excusa para construir los códigos de Reed-Solomon y los códigos BCH. Consideremos ahora $P \subset \mathbb{F}_q$ un subconjunto de $n \leq q$ elementos. Podemos asociar a $c \in \mathbb{F}_q^k$ el polinomio $c(x) = \phi(c)$ y evaluar este polinomio en los elementos de P.

Definición 3.8. Sean $1 < k < n < q$, y consideremos $P_k = \{f(x) \in \mathbb{F}_q[x] : \deg f \leq k\} \cong \mathbb{F}_q^{k+1}$. Para un conjunto $P = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{F}_q$ de n elementos, definimos el **código de Reed-Solomon** $RS(n, k)$ como

$$RS(n, k) = \{f(x_1), f(x_2), \ldots, f(x_n)) : f \in P_{k-1}\},$$

el cual es un (n, k, d) código lineal. Notar que P no está explícito, y que $c_i = f(x_i)$.

Ejemplo 3.9. Supongamos $q = 7$, $n = 6$ y $k = 4$. Luego tenemos $P_3 = \{ax^3 + bx^2 + cx + d : \ a, b, c, d \in \mathbb{F}_7\}$. Como $k = 4$ tenemos que el código estará generado por 4 vectores. Podemos tomar arbitrariamente $P = \{0, 1, 2, 4, 5, 6\}$ por lo que $RS(n, 3)$ estará generado por los vectores

$$(0, 1^3, 2^3, 4^3, 5^3, 6^3), (0, 1^2, 2^2, 4^2, 5^2, 6^2), (0, 1, 2, 4, 5, 6), (1, 1, 1, 1, 1, 1),$$

esto es hay $7^4 = 2401$ palabras generadas por los vectores

$$(0, 1, 1, 1, 6, 6), (0, 1, 4, 2, 4, 1), (0, 1, 2, 4, 5, 6), (1, 1, 1, 1, 1, 1).$$

En particular vemos que $d \leq 3$ ya que restando dos vectores vemos $(6, 0, 0, 0, 5, 5) \in RS(n, 3)$. ¿Será $d = 2$ ó $d = 1$?

Teorema 3.10. *$RS(n, k)$ es un (n, k, d) código lineal con $d = n+1-k$. Una matriz generadora es una matriz de tipo Vandermonde de $k \times n$*

$$G = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{k-1} & x_2^{k-1} & \cdots & x_n^{k-1} \end{pmatrix}.$$

Demostración. Es claro que $n = n$ y que $k = k$. Veamos que $d = n + 1 - k$. Recordando que $d = \text{mín}\{w(c) : c \in RS(n, k)\}$ podemos buscar el peso mínimo. Pero como toda palabra c está asociada a una función f, vemos que $c_i = 0 \iff f(x_i) = 0$. Como $\deg f \leq k - 1$ tenemos que f posee a lo sumo $k - 1$ ceros. Luego al menos $n - (k - 1) = n + 1 - k$ elementos $x_i \in P$ no son ceros de f, por lo que toda c posee al menos $n + 1 - k$ coordenadas no nulas. □

Notemos que podemos hablar de un código de Reed-Solomon como $RS(n, k)$ ó como $RS(n, n-r)$ donde su distancia mínima es $d = r+1$. A partir de ahora usaremos $P = \alpha, \alpha^2, \ldots, \alpha^{q-1}$. Para elegir convenientemente este conjunto necesitamos

Definición 3.11. Sea $\mathbb{F}_q$ un cuerpo finito. Una **raíz primitiva** de $\mathbb{F}_q$ es un elemento $0 \neq \zeta \in \mathbb{F}_q$ tal que

$$\mathbb{F}_q = \{0, \zeta, \zeta^2, \zeta^3, \zeta^4, \ldots, \zeta^{q-1}\},$$

donde en particular $\zeta^q = \zeta$ y $\zeta^{q-1} = 1$

Proposición 3.12. *Sean $n = q-1$ y $r \geq 1$. Todo código de Reed-Solomon $RS(n, n-r)$ es cíclico con generador $g(x) = (x-\alpha)(x-\alpha^2)\cdots(x-\alpha^r)$ donde estamos tomando el producto en $\mathbb{F}_q[x]$, y α es una raíz primitiva de $\mathbb{F}_q$*

Demostración. Notemos que $\mathbb{F}_q[x]$ es un dominio de ideales principales, por lo cual la factorización es única y este elemento es el mínimo común múltiplo de los monomios. Por otro lado $\mathbb{F}_q^*$ es un grupo multiplicativo de tamaño $q-1$ por lo que todo $a \in \mathbb{F}_q$ verifica $a^{q-1} = 1$ y por tanto $x^n - 1 = x^{q-1} = \prod(x-a) = \prod_{i=1}^{q-1}(x-\alpha^i)$. De este modo $g(x) \mid x^n - 1$ es un producto de factores irreducibles. El código cíclico C generado por $g(x)$ es un (n, k, d) código con $k = n-r$ y $d \leq r$. Veamos que $RS(n, n-r) = C$, a lo cual es suficiente ver $RS(n, n-r) \subset C$ ya que ambos tienen dimensión $n-r$.

Sea $c \in \mathbb{F}_q^n$ con $\phi(c) = c_0 + c_1 x + \ldots + c_{n-1}x^{n-1}$. Si $c \in RS(n, n-r)$ es $c_j = f(\alpha^j)$ para $f(x) = \sum_{m=0}^{n-r} f_m x^m$. Luego

$$\phi(c) = \sum_{i=0}^{n-1}\sum_{m=0}^{n-r} f_m \alpha^{mi} x^i$$

Para ver que $c \in C$ basta ver que $\phi(c) = p(x)g(x)$ para algún polinomio $p(x)$, y por el teorema del resto basta ver que $\phi(c)(\alpha^j) = 0$ para $1 \leq j < r$. Luego

$$\begin{aligned}
\phi(c)(\alpha^j) &= \sum_{i=0}^{n-1}\sum_{m=0}^{n-r} f_m \alpha^{mi}\alpha^{ij} \\
&= \sum_{m=0}^{n-r}\sum_{i=0}^{n-1} f_m \alpha^{mi}\alpha^{ij} \\
&= \sum_{m=0}^{n-r} f_m \sum_{i=0}^{n-1} \alpha^{(m+j)i} \\
&= \sum_{m=0}^{n-r} f_m \left(\frac{\alpha^{(m+j)n} - 1}{\alpha^{(m+j)} - 1}\right) \\
&= \sum_{m=0}^{n-r} f_m \left(\frac{(\alpha^{q-1})^{(m+j)} - 1}{\alpha^{(m+j)} - 1}\right)
\end{aligned}$$

Como $0 \leq m \leq n-r$ y $1 \leq j < r$ es $1 \leq j \leq n-1$ y $\alpha^{m+j} \neq 0$. Luego $\phi(c)(\alpha^j) = \sum f_m \cdot 0 = 0$ por lo que $c \in C$ como queríamos ver. □

Definición 3.13. Sean $n \geq 0$ y $t \geq 1$ el menor natural tal que $n \mid q^t - 1$. Un **coset** $q-$**ciclotómico** $C_q(a)$ es el conjunto $\{a, qa, q^2a, \ldots, q^{t-1}a\} \pmod{n}$, el cual tiene $\delta(a) \mid t$ elementos.

Ejemplo 3.14. Si $q = 3$ y $n = 13$ entonces $t = 3$ y los cosets $3-$ciclotómicos son (0), (1,3,9), (2,6,5), (4,12,10), (7,8,11). Si $q = n + 1$ entonces los cosets $q-$ciclotómicos son $C_q(a) = (a)$.

Definición 3.15. Sean $n \geq 0$ y $t \geq 1$ como antes. Sea α una raíz primitiva de $\mathbb{F}_{q^t}$ y $1 \leq r \leq n$. El **polinomio minimal** de α^i es

$$m(\alpha^i) = \prod_{j \in C_q(i)} (x - \alpha^j) \in \mathbb{F}_q[x]$$

Lo cual nos permite ver en $\mathbb{F}_q[x]$ el polinomio $(x - \alpha) \in \mathbb{F}_{q^t}[x]$.

Ejemplo 3.16. Usando que $256 = 2^8$, podemos asociar a cada **byte** $x \in \mathbb{F}_{256} = \mathbb{F}_2(\alpha)$ un vector de 8 **bits** $(b_0, \ldots, b_8) \in \mathbb{F}_2^8$. Un código de Reed-Solomon $RS(n, k)$ puede utilizarse para codificar un mensaje en un código QR de la siguiente manera:

1. Se determinan parámetros como

   - el largo máximo L del mensaje
   - el nivel de corrección de errores $0 < t < 50\,\%$ que se quiere asegurar.

   aquí queda determinado unívocamente cuál será el código $RS(n, k)$ a utilizar y el tamaño de nuestro código QR

2. Escribir el mensaje (de longitud $N \leq L$) con caracteres en formato `ascii`,

3. Asignar cada carácter a su respectivo número $\texttt{1} \leq \texttt{a} \leq \texttt{255}$.

4. Escribir cada byte como una sucesión de 8 bits.

5. Codificar los parámetros del código QR (que eran L, R) con 20 bits.

6. Concatenar todo el mensaje y formar una única tira de longitud $20 + 8N$.

7. Dividirla en bloques B_i de longitud k_i adecuada añadiendo 0 si es necesario.

8. Cada bloque B_i se codifica con un código $RS(n, k_i)$. Se eliminan los primeros bits y se obtiene un nuevo bloque B_i' de longitud $(2t + 1)k_i$

9. Concatenar todos los bloques y colocarlo en el código QR a construir
10. Aplicar alguna transformación para que sea más heterogéneo

Bibliografía Sugerida

Por sección:

- O. Pretzel, "Error-Correcting Codes and Finite Fields", Oxford, 1992
- N. Biggs, "Discrete Mathematics", Oxford, 1998
- Huffman and Pless, "Fundamentals of Error-Correcting Codes", Cambridge, 2003

Índice Alfabético