UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA

LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

Tema 6. Códigos Lineales

José A. Montenegro

Dpto. Lenguajes y Ciencias de la Computación
ETSI Informática. Universidad de Málaga
monte@lcc.uma.es twitter

26 de septiembre de 2013

Introducción a códigos lineales

- Las técnicas para construir códigos útiles pueden ser extendidas enormemente si dotamos a los símbolos con las propiedades de los números.
- Los códigos 'aritméticos'para la compresión de datos descritos anteriormente son un ejemplo.
- Ahora utilizaremos métodos algebraicos para construir códigos con el propósito de corregir los datos.
- En el caso de un alfabeto binario $\mathbb{B}$, los símbolos 0 y 1 pueden ser sumados y multiplicados acorde las siguientes reglas:

$$0+0=0, 0+1=1, 1+0=1, 1+1=0;$$

$$0\times 0=0, 0\times 1=0, 1\times 0=0, 1\times 1=1.$$

- Utilizando la nomenclatura de Algebra, establecemos que el conjunto $\mathbb{B}$, con estas operaciones es un *Cuerpo*, anotado como $\mathbb{F}_2$.
- El método más útil para construir códigos binarios depende del hecho que el conjunto $\mathbb{F}_2^n$ de palabras de longitud n en $\mathbb{F}_2$ es un *espacio de vectores*.
- Las reglas para la suma de vectores y la multiplicación escalar son las siguientes:

$$(x_1 x_2 \dots x_n) + (y_1 y_2 \dots y_n) = (x_1 + y_1 \ x_2 + y_2 \dots x_n + y_n);$$

$$\begin{aligned} 0 \times (x_1 x_2 \dots x_n) &= (00 \dots 0), \\ 1 \times (x_1 x_2 \dots x_n) &= (x_1 x_2 \dots x_n). \end{aligned}$$

- Un subconjunto C del espacio de vectores $\mathbb{F}_2^n$ es un subespacio si cualquier $x, y \in C$ cumple $x + y \in C$.

Definición 1 (Código Lineal)

Un código lineal (binario) es un subespacio C del espacio de vectores $\mathbb{F}_2^n$.

- Por ejemplo, el subconjunto $C = \{000, 110, 011\}$ de $\mathbb{F}_2^3$ no es un código lineal, debido a que $110 + 011 = 101$, el cual no está en C.
- Por otro lado, el código repetición $R_n \subseteq \mathbb{F}_2^n$, contiene las dos palabras $000\ldots00$ y $111\ldots11$ es un código lineal para cualquier n, ya que $111\ldots11 + 111\ldots11 = 000\ldots00$.

- Hemos visto anteriormente que la construcción de buenos códigos requiere un compromiso entre la tasa de información ρ y la distancia mínima δ. En el caso de códigos lineales podemos ser más específicos con estos parámetros.
- Ya que un código lineal C es un subespacio de $\mathbb{F}_2^n$ tiene una dimensión k, definida como el tamaño del conjunto mínimo de expasión (bases).
- Cada elemento de C puede ser expresado de forma única como una combinación lineal de la bases, por lo que $|C| = 2^k$ para algún k en el rango $0 \leq k \leq n$.
- La tasa de información de C es

$$\rho = \frac{log_2|C|}{n} = \frac{k}{n}$$

- Entonces es conveniente utilizar k en vez de ρ y usualmente describiremos un código lineal por los parámetros (n, k, δ).
- Para los códigos en general, encontrar δ es tedioso, ya que requiere la comparación de todos los pares de palabras codificadas, pero para un código lineal existe una forma relativamente fácil.

Definición 2 (Peso)

El peso w(x) de una palabra $x \in \mathbb{F}_2^n$ es el número de 1s en x.
Equivalentemente, w(x)=d(x,0), donde 0 denota la palabra de todos los ceros $000\ldots000$

Lema 1

Para un código lineal, la mínima distancia δ es igual al mínimo peso de una una palabra codificada que no es $000\ldots000$.

Ejemplo 1

Establezca los parámetros (n, k, δ) *de los siguientes códigos lineales.*

$$D_1 = \{000000, 100000, 010000, 110000\},$$
$$D_2 = \{000000, 111000, 000111, 111111\}.$$

Ejemplo 1

Establezca los parámetros (n, k, δ) *de los siguientes códigos lineales.*

$$D_1 = \{000000, 100000, 010000, 110000\},$$
$$D_2 = \{000000, 111000, 000111, 111111\}.$$

Solución:

1. D_1 tiene palabra de longitud n=6 y ya que hay $4 = 2^2$ palabras codificadas, la dimensión es k=2 (y la tasa es $\rho = 1/3$).
   - El peso de las palabras codificadas distintas de cero es 1,1,2 por lo que $\delta = 1$.
   - Con este código, no es posible la corrección de errores.
2. D_2 también tiene dimensión y tasa $1/3$, pero la distancia mínima es $\delta = 3$, y es un código de corrección de errores de 1 bit.

- La construcción de los códigos lineales C con los valores de los parámetros (n, k, δ) está limitado por el límite de embalaje.
- Cuando $|C| = 2^k$, y $\delta \geq 2r + 1$ (por lo que C es un código de error r) la condición es

$$2^k \left(1 + \left(\begin{array}{c} n \\ 1 \end{array} \right) + \left(\begin{array}{c} n \\ 2 \end{array} \right) + \ldots + \left(\begin{array}{c} n \\ r \end{array} \right)\right) \leq 2^n$$

Ejemplo 2

Establezca un límite superior para el número de palabras codificadas en un código lineal con longitud de palabra 12, si necesitamos un código corrector error 2.

Solución:

- El límite de embalaje es $2^k(1 + 12 + 66) \leq 4096$, por lo que $2^k \leq 50$ aproximadamente.
- Ya que k debe ser un entero el máximo posible de 2^k es $2^5 = 32$.
- (Aún no hemos determinado como construir tal código).

Ejercicio 1

¿Cual de los siguientes subconjuntos de $\mathbb{F}_2^3$ es un código lineal?
$C_1 = \{000, 110, 101, 011\}$, $C_2 = \{000, 100, 010, 001\}$.

Ejercicio 1

¿Cual de los siguientes subconjuntos de $\mathbb{F}_2^3$ es un código lineal?
$C_1 = \{000, 110, 101, 011\}$, $C_2 = \{000, 100, 010, 001\}$.

Solución:

C_1 110+101= 011, 110+011= 101, 101+011= 110.

C_2 100+010= 110 No pertenece a C_2.

Ejercicio 2

Supongamos que queremos enviar uno entre 128 mensajes diferentes, y cada mensaje esta representado por una palabra codificada de longitud 10. ¿Es posible construir un código linear corrector-1 que satisface estas condiciones?

Ejercicio 2

Supongamos que queremos enviar uno entre 128 mensajes diferentes, y cada mensaje esta representado por una palabra codificada de longitud 10. ¿Es posible construir un código linear corrector-1 que satisface estas condiciones?

Solución:
n= 10, k= 7
Según el límite de embalaje tenemos $2^k(1+n) \leq 2^n$, $128(1+10) \leq 1024$ No se cumple

Ejercicio 3

Es necesario asignar a cada estudiante un número identificativo mediante una palabra binaria. (a) Si hay 53 estudiantes, encuentra menor dimensión posible de un código lineal para este propósito. (b) Si el código debe permitir la corrección de 1 error, encontrar la menor longitud posible de las palabras codificadas.

Ejercicio 3

Es necesario asignar a cada estudiante un número identificativo mediante una palabra binaria. (a) Si hay 53 estudiantes, encuentra menor dimensión posible de un código lineal para este propósito. (b) Si el código debe permitir la corrección de 1 error, encontrar la menor longitud posible de las palabras codificadas.

Solución:

a) $53{,}2^6$

b) $2^6(1+n) \leq 2^n; 1+n \leq 2^{(n-6)}; (n=10);$

Ejercicio 4

Supongase que necesitamos construir un código lineal con n=12 y $\delta = 3$. Encuentra un límite superior para la tasa de información del código mencionado.

Ejercicio 4

Supongase que necesitamos construir un código lineal con n=12 y $\delta = 3$. Encuentra un límite superior para la tasa de información del código mencionado.

Solución:

$2^k(1+12) \leq 2^{12}; 2^k \leq 2^{12}/13; 2^k \leq 315; k = 8$

$\rho = k/n; \rho = 8/12, \rho = 2/3$

Ejercicio 5

Sea $B(n, \delta)$ que denota la máxima dimensión de un código lineal en $\mathbb{F}_2^n$ con mínima distancia δ. Haciendo uso del límite de embalaje mostrar que $B(6, 3) \leq 3$, $B(7, 3) \leq 4$, $B(8, 3) \leq 4$.

Ejercicio 5

Sea $B(n, \delta)$ que denota la máxima dimensión de un código lineal en $\mathbb{F}_2^n$ con mínima distancia δ. Haciendo uso del límite de embalaje mostrar que $B(6, 3) \leq 3$, $B(7, 3) \leq 4$, $B(8, 3) \leq 4$.

Solución:

$2^k(1+6) \leq 2^6; 2^k \leq 9,14. k = 3$

$2^k(1+7) \leq 2^7; 2^k \leq 16. k = 4$

$2^k(1+8) \leq 2^8; 2^k \leq 28{,}4. k = 4$

Construcción de códigos lineales utilizando matrices

- Anteriormente hemos estudiado el problema de transmitir información según una tasa establecida, mientras reducimos la probabilidad de confusión.
- La idea principal es dividir el flujo original de bits en bloques de tamaño k y asignar una palabra codificada de longitud n a cada uno de los 2^k posibles bloques.
- O en otras palabras, necesitamos una función de codificación $\mathbb{F}_2^k \to \mathbb{F}_2^n$.
- Dado que $\mathbb{F}_2^k$ y $\mathbb{F}_2^n$ son espacio de vectores, un candidato para tal función es una transformación lineal, definida por una matriz.

- En el contexto del Algebra matricial es a menudo conveniente sustituir una palabra x, considerada como un vector fila, por el correspondiente vector columna x'(la transpuesta de x).
- Por tanto, si E es una $n \times k$ matriz en $\mathbb{F}_2$ e y en $\mathbb{F}_2^k$ entonces la palabra x definida por $x' = Ey'$ está en $\mathbb{F}_2^n$.
- La matriz E define una función de codificación $\mathbb{F}_2^k \to \mathbb{F}_2^n$.

Ejemplo 3

En el ejemplo del tema anterior establecimos una regla que asignaba a cada bloque $y_1y_2y_3$ *de tamaño 3 una palabra codificada* $x_1x_2x_3x_4x_5x_6$ *de tamaño 6. Expresa esta regla en forma de matriz.*

$$x_1 = y_1$$
$$x_2 = y_2$$
$$x_3 = y_3$$
$$x_4 = 0 \text{ si } y_1 = y_2, \text{ si no } x_4 = 1$$
$$x_5 = 0 \text{ si } y_2 = y_3, \text{ si no } x_5 = 1$$
$$x_6 = 0 \text{ si } y_1 = y_3, \text{ si no } x_6 = 1$$

Solución:
La regla para x_4 es $x_4 = 0$ si $y_1 = y_2$, si no $x_4 = 1$. Utilizando la estructura algebraica del cuerpo $\mathbb{F}_2$ podemos expresarlos mediante una ecuación lineal $x_4 = y_1 + y_2$. De las misma forma tenemos $x_5 = y_2 + y_3$ y $x_6 = y_1 + y_3$. Todas las ecuaciones pueden ser escritas mediante una simple ecuación de matrices.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_1 + y_2 \\ y_2 + y_3 \\ y_1 + y_3 \end{pmatrix} = E \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

donde E es la matriz 6 x 3 siguiente:

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

Como anteriormente, podemos verifica que el código resultante $C \subseteq \mathbb{F}_2^6$ tiene los parámetros n $= 6$, k $= 3$, and $\delta = 3$, y por tanto es un código corrector de error 1 con una tasa $1/2$.

- En general, dado cualquier matriz E (n x k) sobre $\mathbb{F}_2$, sea C el conjunto

$$\{x \in \mathbb{F}_2^n \mid \text{x'=E y'para algún } y \in \mathbb{F}_2^k\}.$$

- Si x y w pertenecen a C, tenemos que $x' = Ey'$ y $w' = Ez'$, entonces (x+w)'=E (y+z)', por lo que x + w también están en C. Por tanto C es un código lineal.
- Esencialmente, ahora tenemos la respuesta al problema de codificar un flujo de bits. El flujo es dividido en bloques y de longitud k, y la palabras codificada para y es x, donde $x' = Ey'$, para una matriz E apropiada.
- Debemos intentar asegurar que el código resultante tiene buenas propiedades de corrección de errores, y que existe un método para implementar la regla MD, que veremos a continuación.

Ejercicio 6

Supongamos que codificamos un bloque de bits $y_1 y_2 \dots y_k$ *estableciendo* $x_i = y_i$ *para* $i = 1, 2, \dots k$ *y definimos el bit de verificación de paridad* x_{k+1} *como sigue:* $x_{k+1} = 0$ *si hay un número par de yi's=1,* $x_{k+1} = 1$ *si el número de yi's=1 son impar. Escribir la matriz E tal que x'= E y'.*

Ejercicio 6

Supongamos que codificamos un bloque de bits $y_1 y_2 \dots y_k$ estableciendo $x_i = y_i$ para $i = 1, 2, \dots k$ y definimos el bit de verificación de paridad x_{k+1} como sigue: $x_{k+1} = 0$ si hay un número par de yi's=1, $x_{k+1} = 1$ si el número de yi's=1 son impar. Escribir la matriz E tal que x'= E y'.

Solución:
La matriz sería la identidad para que los $x_1 x_2 \dots y_k$ y para x_{k+1} sería toda la fila a 1.
Para el caso de k=3 sería:

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

Ejercicio 7

Mostrar que en el código definido en el ejercicio anterior cada palabra tiene un peso par.

Ejercicio 7

Mostrar que en el código definido en el ejercicio anterior cada palabra tiene un peso par.

Solución:
Tenemos solamente dos casos.
a) La palabra tiene un peso par de 1 se añade un 0 en el código de control, con lo cual la palabra será par.
b) La palabra tiene un peso impar con lo cual se añade un 1 como código de control, y la palabra vuelve a ser par.

Ejercicio 8

Establece la tasa de información ρ y la mínima distancia δ para el código de verificación de paridad.

Ejercicio 8

Establece la tasa de información ρ y la mínima distancia δ para el código de verificación de paridad.

Solución:
$\rho = \frac{k}{k+1}$ $\delta = 2$. Mínimo par por el peso son 2.

La matriz de verificación de un código lineal

- Anteriormente, hemos descrito un método para codificar un flujo de bits dividiendo en bloques de longitud k y aplicando una matriz E $(n \times k)$ para cada bloque y.
- En el ejemplo 3 la transformación $x' = Ey'$ estaba definida por las siguientes ecuaciones lineales:

$$x_1 = y_1,\ x_2 = y_2,\ x_3 = y_3,$$
$$x_4 = y_1 + y_2,\ x_5 = y_2 + y_3,\ x_6 = y_1 + y_3.$$

- Las variables y_1, y_2, y_3 pueden ser eliminadas de la ecuación, obteniendo

$$x_1 + x_2 + x_4 = 0,\ x_2 + x_3 + x_5 = 0,\ x_1 + x_3 + x_6 = 0.$$

Nota: -1 = 1 en el cuerpo $\mathbb{F}_2$.

- Estas ecuaciones pueden ser reescritas de la forma de una ecuación de matrices $Hx' = 0'$, donde

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

- Por tanto tenemos una forma alternativa de definir el código, como el conjunto de x tal que $Hx' = 0'$, que es un subespacio de $\mathbb{F}_2^n$ y por tanto un código lineal.
- Podemos determinar el caso general en el siguiente lema.

Lema 2

Sea E una matriz n x k sobre $\mathbb{F}_2$, de la forma

$$\begin{pmatrix} I \\ A \end{pmatrix}$$

- *donde I es la matriz identidad con tamaño k, y A es cualquier matriz $(n-k) \times k$.*
- *Entonces el código $\{x \in \mathbb{F}_2^n \mid$ x'=E y'para algún $y \in \mathbb{F}_2^k\}$.*
- *puede ser definido como $\{x \in \mathbb{F}_2^n \mid$ H x'=0'$\}$.*
- *donde H es la $(n-k) \times n$ matriz (A I). (I es la matriz identidad con tamaño $n-k$.)*

Definición 3 (Matriz Verificación)

Una matriz H en $\mathbb{F}_2$ con m filas y n columnas es la matriz de verificación para el código lineal

$$C = \{x \in F_2^n | Hx' = 0'\}.$$

En la definición H puede ser cualquier matriz m x n sobre $\mathbb{F}_2$. Tal y como hemos definido H tiene la forma estándar (AI), donde A tiene n-m columnas y I tiene m columnas. Entonces el código correspondiente tiene dimensiones k = n - m. Para probar esto, tenemos

$$H = (AI) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2k} & 0 & 1 & \dots & 0 \\ . & . & \dots & . & . & . & \dots & . \\ . & . & \dots & . & . & . & \dots & . \\ a_{m1} & a_{m2} & \dots & a_{mk} & 0 & 0 & \dots & 1 \end{pmatrix}$$

Por tanto las ecuaciones $Hx' = 0'$ son

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k + x_{k+1} &= 0 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k + x_{k+2} &= 0 \\ \ldots\ldots & \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mk}x_k + x_n &= 0 \end{aligned}$$

Estas ecuaciones pueden ser reordenadas de forma que los valores $x_{k+1}, x_{k+2}, \ldots x_n$ son definidos en términos de los valores $x_1, x_2, \ldots, x_k$:

$$\begin{aligned} x_{k+1} &= a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k \\ x_{k+2} &= a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k \\ \ldots\ldots & \\ x_{k+2} &= a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mk}x_k \end{aligned}$$

- En una palabra codificada $x = x_1 x_2 \dots x_n$, estableceremos que

$$x_1, x_2, \dots, x_k \text{ son los bits de los mensajes}$$
$$x_{k+1}, x_{k+2}, \dots, x_n \text{ son los bits de verificación.}$$

- Si los bits de mensajes tienen asignados valores arbitrarios, los valores de los bits de verificación son determinados por las ecuaciones anteriormente explicados.
- Ya que tenemos 2^k posibles valores de mensajes, la dimensión del código es k.

Ejemplo 4

Establezca una lista de las palabras codificadas definidas por la matriz de verificación

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

¿Cuales son los parámetros de este código?

Solución:
Si $x = x_1x_2x_3x_4$ entonces la condición $Hx' = 0'$ tenemos que

$$x_1 + x_3 = 0,\ x_1 + x_2 + x_4 = 0.$$

Podemos reescribir las ecuaciones de forma que x_3 y x_4 son dados en términos de x_1 y x_2:

$$x_3 = x_1,\ x_4 = x_1 + x_2.$$

Las palabras codificadas pueden ser encontradas dando todos los posibles valores a x_1, x_2 y utilizando las ecuaciones para calcular x_3 y x_4.

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & peso \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 \\ 1 & 0 & 1 & 1 & 3 \\ 1 & 1 & 1 & 0 & 3 \end{pmatrix}$$

El código tiene parámetros n=4, k=2, y la mínima distancia es igual al mínimo peso que no es cero, el cual es $\delta = 2$.

Ejercicio 9

Establece una lista de palabras codificadas definidas por la matriz de verificación

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

¿Cuales son los parámetros de este código?

Solución:
Si $x = x_1x_2x_3x_4$ entonces la condición $Hx' = 0'$ tenemos que

$$x_1 + x_3 = 0,\ x_1 + x_2 + x_4 = 0,\ x_2 + x_5 = 0.$$

Podemos reescribir las ecuaciones de forma que x_3 x_4 y x_5 son dados en términos de x_1 y x_2:

$$x_3 = x_1,\ x_4 = x_1 + x_2,\ x_5 = x_2$$

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & peso \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 3 \\ 1 & 0 & 1 & 1 & 0 & 3 \\ 1 & 1 & 1 & 0 & 1 & 4 \end{pmatrix}$$

El código tiene parámetros n=5, k=2, y la mínima distancia es igual al mínimo peso que no es cero, el cual es $\delta = 3$.

Ejercicio 10

El código $C_1 = \{00, 01, 10, 11\}$ *pude ser usado para representar 4 mensajes* $\{a, b, c, d\}$. *El código* C_r *es obtenido mediante repetición de cada palabra en* C_1 r *veces: por ejemplo, en* C_3 *el mensaje b es representado por 010101. Muestra que* C_r *es un código lineal y establece sus parámetros* (n, k, δ) *y la matriz de verificación asociada.*

Ejercicio 10

El código $C_1 = \{00, 01, 10, 11\}$ pude ser usado para representar 4 mensajes $\{a, b, c, d\}$. El código C_r es obtenido mediante repetición de cada palabra en C_1 r veces: por ejemplo, en C_3 el mensaje b es representado por 010101. Muestra que C_r es un código lineal y establece sus parámetros (n, k, δ) y la matriz de verificación asociada.

Solución:
Ya que C_1 es lineal por extensión podemos determinar que C_r es lineal.
Los parámetros son n $= 2$r,k$=2$,$\delta = r$.
La matriz de verificación representa estas ecuaciones $x_1 = x_3 = x_5 = \ldots$ y $x_2 = x_4 = x_6 = \ldots$

Construyendo códigos correctores 1 bit

El siguiente teorema describe una forma muy fácil para construir una matriz de verificación para un código corrector 1 bit.

Teorema 1

El Código C definido por una matriz de verificación H es un código corrector 1 cumpliendo que

1. *No existe una columna en H con todos los elementos 0s.*
2. *No tenemos dos columnas de H idénticas.*

Ejemplo 5

En un ejemplo anterior mostramos que el menor valor posible de n y k para un código corrector 1 con tasa de información 0.8 son n=25, k=20. ¿Como podemos construir el código utilizando una matriz de verificación?

Ejemplo 5

En un ejemplo anterior mostramos que el menor valor posible de n y k para un código corrector 1 con tasa de información 0.8 son n=25, k=20. ¿Como podemos construir el código utilizando una matriz de verificación?

Solución:

- H es una matriz m x n con m=n-k $= 5$ filas.
- Acorde con el teorema 1, las n$= 25$ columnas deben ser distintas, y ninguna columna puede ser cero.
- Concretamente hay $2^5 - 1 = 31$ posibles columnas y cualquiera 25 de esas columnas puedan ser elegidas.
- Si H esta en forma estándar las últimas cinco columnas serán 5 palabras de peso 1,
- y las primeras 20 columnas pueden ser cualquiera palabra que no sea cero y distintas a las 5 últimas columnas.

Teorema 2

Sea $C \subseteq \mathbb{F}_2^n$ un código lineal definido por una matriz de verificación H.
Suponemos que un error en un bit sucede en la transmisión de una palabra codificada (palabra recibida es z).
El error ocurre en el i^{th} bit de z, donde i es determinado por el hecho que Hz' es igual a la i^{th} columna de H.

- Asumiendo que no tenemos más de un bit de error en cada palabra codificada, podemos hacer uso del procedimiento mostrado en la figura 1.

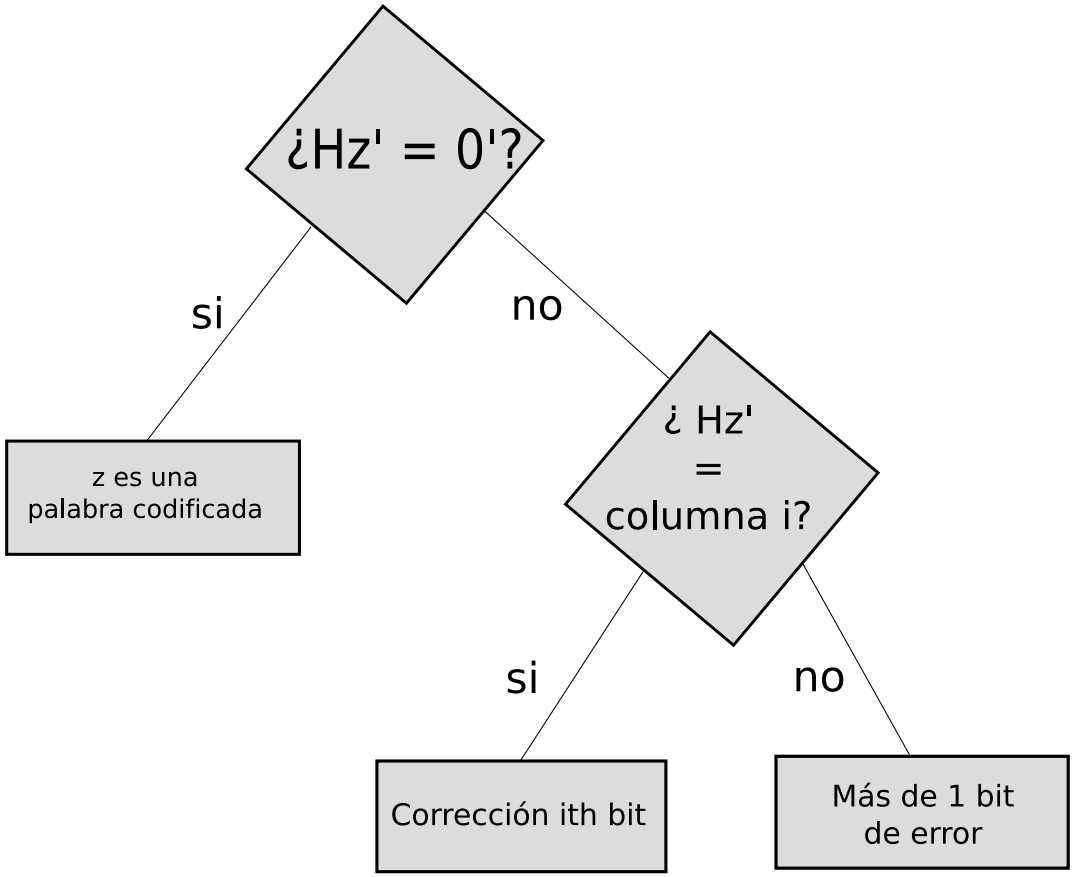

Figura 1 : Procesando una palabra recibida z para un código con matriz de verificación H

- Para cada palabra recibida z el Receptor deber calcular Hz'.
- Si $Hz' = 0$, entonces z es la palabra codificada correcta.
- Si Hz' es igual a la columna i^{th} de H, entonces la palabra codificada correcta es obtenida cambiando el i^{th} bit en z.
- En otro caso, al menos dos bit son erróneos.

Ejercicio 11

Construya un Código Corrector 1 $C \in \mathbb{F}_2^6$ *con* $|C| = 8$.

Ejercicio 11

Construya un Código Corrector 1 $C \in \mathbb{F}_2^6$ con $|C| = 8$.

Solución:

H es una matriz $m \times n$ con m=n-k.

Del enunciado sabemos que n=6 y k=3, por lo que debemos construir una matriz 3×6.

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & peso \\ 1 & 1 & 1 & 1 & 0 & 0 & 4 \\ 1 & 1 & 0 & 0 & 1 & 0 & 3 \\ 1 & 0 & 1 & 0 & 0 & 1 & 3 \end{pmatrix}$$

Ejercicio 12

Establezca todas las palabras codificadas que pertenecen a un código lineal con la siguiente matriz de verificación, y encuentra los parámetros $(n, k\delta)$ para este código.

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

Solución:
La matriz identidad está formada por las columnas 2,5,3 y 6. Esto significa que x_2, x_5, x_3 y x_6 pueden ser determinadas por x_1, x_4 y x_7. Explícitamente podemos determinar estas ecuaciones de la forma:

$$\begin{aligned} x_2 &= x_1 + x_4 + x_7 \\ x_5 &= x_4 + x_7 \\ x_3 &= x_1 + x_4 + x_7 \\ x_6 &= x_7 \end{aligned}$$

Por tanto, tenemos 8 palabras codificadas determinadas por los bits x_1, x_4 y x_7 (2^3).

$$\begin{aligned} x_1 = 0, x_4 = 0, x_7 = 0 &\text{ tendremos } x_2 = 0, x_3 = 0, x_5 = 0, x_6 = 0 \mapsto 0000000 \\ x_1 = 0, x_4 = 0, x_7 = 1 &\text{ tendremos } x_2 = 1, x_3 = 1, x_5 = 1, x_6 = 1 \mapsto 0110111 \\ x_1 = 0, x_4 = 1, x_7 = 0 &\text{ tendremos } x_2 = 1, x_3 = 1, x_5 = 1, x_6 = 0 \mapsto 0111100 \\ x_1 = 0, x_4 = 1, x_7 = 1 &\text{ tendremos } x_2 = 0, x_3 = 0, x_5 = 0, x_6 = 1 \mapsto 0001011 \\ x_1 = 1, x_4 = 0, x_7 = 0 &\text{ tendremos } x_2 = 1, x_3 = 1, x_5 = 0, x_6 = 0 \mapsto 1110000 \\ x_1 = 1, x_4 = 0, x_7 = 1 &\text{ tendremos } x_2 = 0, x_3 = 0, x_5 = 1, x_6 = 1 \mapsto 1000111 \\ x_1 = 1, x_4 = 1, x_7 = 0 &\text{ tendremos } x_2 = 0, x_3 = 0, x_5 = 1, x_6 = 0 \mapsto 1001100 \\ x_1 = 1, x_4 = 1, x_7 = 1 &\text{ tendremos } x_2 = 1, x_3 = 1, x_5 = 0, x_6 = 1 \mapsto 1111011 \end{aligned}$$

Sabemos n=7, k=3 y $\delta = 3$ que es la mínima distancia posible para código corrector 1.

Ejercicio 13

Una palabra codificada del código definido por la siguiente matriz ha sido enviada, y la palabra 111010 ha sido recibida.

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

¿Cual es la palabra codificada, asumiendo que solamente un error ha ocurrido?

Ejercicio 13

Una palabra codificada del código definido por la siguiente matriz ha sido enviada, y la palabra 111010 ha sido recibida.

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

¿Cual es la palabra codificada, asumiendo que solamente un error ha ocurrido?

Solución:
Dada z'=[111010]', encontramos Hz'=[110]', que coincide con la 2 columna de H, por lo que el error ha sido en el 2 bit.
La palabra correcta es c=101010 y ahora Hc=0'.

Ejercicio 14

En el ejercicio anterior, ¿Cuantas palabras no pueden ser corregidas, bajo la suposición que al menos un error ha ocurrido?

Ejercicio 14

En el ejercicio anterior, ¿Cuantas palabras no pueden ser corregidas, bajo la suposición que al menos un error ha ocurrido?

Solución:

- n= 6, por lo que tenemos $2^6 = 64$ palabras y $2^3 = 8$ palabras codificadas.
- Cada $N_1(c)$ contiene 6+1 elementos.
- Por tanto 8 x 7 palabras pueden ser corregidas.
- Por lo que 64 - 56= 8 palabras no pueden ser corregidas.

El problema de la decodificación

- De forma genérica, no hay una forma fácil de implementar la regla de decisión MD. (Dado un conjunto de palabras codificadas $C \subseteq \mathbb{B}^n$ y una palabra $z \in \mathbb{B}^n$, el problema es encontrar una palabra codificada $c \in C$ que sea cercana a z, según de la distancia de Hamming).
- Podemos utilizar el método por fuerza bruta mediante una lista de todas las palabras codificadas, pero no es un método eficiente.
- Cuando C es definido mediante una construcción algebraica, existen maneras de mejorar la propuesta de fuerza bruta.
- Generalmente el problema de decodificación puede ser simplificado utilizando una técnica conocida como *decodificación por síndrome*.

Definición 4 (Síndrome)

- *Sea C el código lineal definido por una matriz de verificación H.*
- *Para cualquier palabra $z \in \mathbb{F}_2^n$ el síndrome de z es s, donde $Hz' = s'$.*
- *Si z es una palabra codificada entonces el síndrome de z es la palabra cero $(s' = 0)$.*
- *Si z es el resultado de realizar un simple error, en el i^{th} de una palabra codificada, entonces el síndrome es la palabra en la i^{th} columna de H.*

Lema 3

Dos palabras $y, z \in \mathbb{F}_2^n$ tiene el mismo síndrome con respecto a C sii $y = z + c$, para algún $c \in C$.

- El conjunto de $y \in \mathbb{F}_2^n$ tal que y=z+c para algún $c \in C$ es conocido como la clase de equivalencia de z con respecto a C y es denotado como z+C.
- La Teoría elemental de Grupo define que dos clases de equivalencia son disjuntos o idénticas (no pueden parcialmente coincidir), por lo que distintas clases de equivalencia forman una partición de $\mathbb{F}_2^n$.

Ejemplo 6

Enumera las clases de equivalencia del código $C \subseteq \mathbb{F}_2^4$ definida por la matriz de verificación.

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

Solución:
En el ejemplo 4 mostraremos que C contiene cuatro palabras codificadas 0000, 0101, 1011, 1110.

Ya que $|C| = 4$ y $|F_2^4| = 2^4 = 16$, hay $16/4 = 4$ clases de equivalencias distintas.

La clase de equivalencia 0000+C es la propia C.

Podemos construir una clase de equivalencia distinta z+C escogiendo cualquier z que no este en la clase de equivalencia, tal que z=1000.

De esta forma, tenemos cuatro clases de equivalencias distintas:

$0000 + C$	$1000 + C$	$0100 + C$	$0010 + C$
0000	1000	0100	0010
0101	1101	0001	0111
1011	0011	1111	1001
1110	0110	1010	1100

- Si el código tiene longitud de palabra n y dimensión k, el número de clases de equivalencias es $2^n/2^k = 2^{n-k} = 2^m$, donde m es el número de filas de H.
- Ya que cada síndrome es una palabra de longitud m, y diferentes clases de equivalencias tienen diferentes síndromes, todas las posibles palabras en $\mathbb{F}_2^m$ se dan como síndromes. Es conveniente ordenarlos en un orden definitivo, como por ejemplo el orden de diccionario.

Definición 5 (Líder Clase de Equivalencia, Tabla de asignación de síndromes)

Un **líder de la clase de equivalencia** *es una palabra de menor peso en su clase de equivalencia, si existe varias posibilidades, escogeremos una de ellas.*

Tabla de asignación de síndromes *es una lista ordenada de pares (s,f) de tal forma que, para cada síndrome s, f es el líder de la clase de equivalencia para la clase de equivalencia que tiene el síndrome s.*

- En la lista de clases de equivalencia del ejemplo 6, el primer elemento es el que tiene menor peso y puede ser escogido como líder.
- Sin embargo, en la tercera clase de equivalencia hay dos palabras de peso 1, y cualquiera de las dos puede ser escogida.
- Si elegimos 0100 de la clase de equivalencia, y ordenamos los síndromes en un orden de diccionario, la *tabla de asignación* de síndrome quedaría de la siguiente forma:

$sindrome$	00	01	10	11
$lider\,clase\,eq.$	0000	0100	0010	1000

El método de decodificación del síndrome está basado en el supuesto que el Receptor conoce que la matriz de verificación H para un código C y tiene una copia de la tabla de asignación.

En ese caso, la siguiente regla de decisión $\sigma : \mathbb{F}_2^n \to C$ puede ser aplicada:

1. Para una palabra recibida z calcular el síndrome s, utilizando la ecuación $s' = Hz'$
2. Establecer el líder de la clase de equivalencia f correspondiente a s
3. Definir $\sigma(z) = z + f$

Teorema 3

Con la notación utilizada anteriormente:

1. $\sigma(z) = z + f$ *es una palabra codificada*
2. *No existe palabra codificada* $c \in C$ *tal que* $d(z, c) < d(z, z + f)$.

La primera parte del teorema muestra que σ es una regla de decisión válida.

La segunda parte muestra que es una regla MD.

La elección del líder de la clase de equivalencia corresponde a escoger uno de las palabras codificadas cercanas a z como la palabra codificada $\sigma(z)$.

Ejemplo 7

La matriz de verificación

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

define el código $C = \{00000, 01011, 10110, 11101\}$.

Construye una tabla de asignación de síndromes para C y utilízalo para determinar $\sigma(10111)$.

Solución:
Las clases de equivalencia pueden ser establecidas de la forma usual.

La primera clase de equivalencia es $00000 + C = C$.

Esta no contiene 10000 (por ejemplo), por lo que la próxima clase de equivalencia es $10000 + C$.

Continuando de esta forma podemos obtener ocho clases de equivalencia

$$00000+C \quad 10000+C \quad 01000+C \quad 00100+C$$
$$00010+C \quad 00001+C \quad 00101+C \quad 00111+C.$$

Tenemos que tener claro que los elementos utilizados para construir las clases de equivalencia no son necesariamente los líderes de las clases de equivalencia.

Para encontrar los líderes de las clases de equivalencia debemos escoger una palabra de cada clase de equivalencia que tenga menor peso.

Por ejemplo, la clase de equivalencia 00111 + C contiene las palabras 00111, 01100 10001, y 11010. Por lo que el líder de la clase de equivalencia puede ser 01100 o 10001.

Supongamos que escogemos 01100 como el líder de esta clase de equivalencia, y realizamos elecciones similares para otras clases de equivalencia.

Para cada líder de las clases de equivalencia f, el síndrome s para todas las palabras de esa clase de equivalencia es dado por $s' = H'f$. Ahora debemos crear la tabla de asignación de síndromes para C según las elecciones realizadas.

000	001	010	011	100	101	110	111
00000	00001	00010	01000	00100	00101	10000	01100

Supongamos que z = 10111 era la palabra recibida.

Ya que $Hz' = 001'$ el correspondiente líder de la clase de equivalencia es f = 00001 y la decisión del Receptor sera que $\sigma(z) = z + f = 10110$.

00000	**00001**	**00010**	00011	00111	**00100**	**00101**	**00110**
01011	01010	01001	**01000**	**01100**	01111	01110	01101
10110	10111	10100	10101	10001	10010	10011	10000
11101	11100	11111	11110	11010	11001	11000	11011

- El método de decodificación por síndrome, no siempre mejora el método de fuerza bruta para implementar la regla MD.
- Si la palabra recibida z es comparada con cada palabra codificada para encontrar cual es la más cercana, entonces, para un código lineal de dimensión k, necesitaremos 2^k comparaciones.
- Con la decodificación por síndrome el Receptor debe tener una copia de la tabla de asignación, que contiene 2^{n-k} entradas, que puede ser considerado como un número grande.
- Afortunadamente, cuando los códigos son construidos utilizando métodos algebraicos hay mejores formas de implementar la regla.

Ejercicio 15

Construir una tabla de asignación de síndromes para el código definido por la matriz de verificación

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

Utilizar la tabla para determinar las palabras codificadas correspondientes a las siguientes palabras: 11111, 11010, 01101, 01110.

Solución:

$x_3 = x_1$

$x_4 = x_2$

$x_5 = x_1 + x_2$

00000	**00001**	**00010**	00011	**00100**	00101	00110	00111
01011	01010	01001	**01000**	01111	01110	01101	**01100**
10101	10100	10111	10110	10001	**10000**	10011	10010
11110	11111	11100	11101	11010	11011	**11000**	11001

sindrome	000	001	010	011	100	101	110	111
lider clase eq.	0000	00001	00010	01000	00100	10000	11000	01100

11111, s =H z= 001, f = 00001, $\sigma(z) = 11111 + 00001 = 11110$

11010, s =H z= 100, f = 00100, $\sigma(z) = 11010 + 00100 = 11110$

01101, s =H z= 110, f = 11000, $\sigma(z) = 01101 + 11000 = 10101$

01110, s =H z= 101, f = 10000, $\sigma(z) = 01110 + 10000 = 11110$

Solución:

	01101	00111
00000	3	3
01011	2	2
10101	2	2
11110	3	3

José A. Montenegro Montes
Dpto. Lenguajes y Ciencias de la Computación
ETSI Informática. Universidad de Málaga
monte@lcc.uma.es
twitter

UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA

LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA